



**LEVERAGING EXTERNAL SENSOR DATA FOR ENHANCED SPACE  
SITUATIONAL AWARENESS**

DISSERTATION

Charlie T. Bellows, Major, USAF

AFIT-ENY-DS-15-S-054

**DEPARTMENT OF THE AIR FORCE  
AIR UNIVERSITY**

***AIR FORCE INSTITUTE OF TECHNOLOGY***

**Wright-Patterson Air Force Base, Ohio**

DISTRIBUTION STATEMENT A: APPROVED FOR PUBLIC RELEASE;  
DISTRIBUTION UNLIMITED

The views expressed in this dissertation are those of the author and do not reflect the official policy or position of the United States Air Force, the Department of Defense, or the United States Government.

This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

AFIT-ENY-DS-15-S-054

LEVERAGING EXTERNAL SENSOR DATA FOR ENHANCED SPACE  
SITUATIONAL AWARENESS

DISSERTATION

Presented to the Faculty  
Dean, Graduate School of Engineering and Management  
Air Force Institute of Technology  
Air University  
Air Education and Training Command  
in Partial Fulfillment of the Requirements for the  
Degree of Doctor of Philosophy

Charlie T. Bellows, BS, MS  
Major, USAF

September 2015

DISTRIBUTION STATEMENT A: APPROVED FOR PUBLIC RELEASE;  
DISTRIBUTION UNLIMITED

AFIT-ENY-DS-15-S-054

LEVERAGING EXTERNAL SENSOR DATA FOR ENHANCED SPACE  
SITUATIONAL AWARENESS

Charlie T. Bellows, BS, MS  
Major, USAF

Committee Membership:

Dr. J. T. Black  
Chair

Dr. R. G. Cobb  
Member

Dr. M. R. Grimaila  
Member

Dr. A. L. Jennings  
Member

ADEDEJI B. BADIRU, PhD  
Dean, Graduate School of Engineering  
and Management

**Abstract**

Reliable Space Situational Awareness (SSA) is a recognized requirement in the current congested, contested, and competitive environment of space operations. A shortage of available sensors and reliable data sources are some current limiting factors for maintaining SSA. Unfortunately, cost constraints prohibit drastically increasing the sensor inventory. Alternative methods are sought to enhance current SSA, including utilizing non-traditional data sources (external sensors) to perform basic SSA catalog maintenance functions. Astronomical data, for example, routinely collects serendipitous satellite streaks in the course of observing deep space; but tactics, techniques, and procedures designed to glean useful information from those collects have yet to be rigorously developed. This work examines the feasibility and utility of performing ephemeris positional updates for a Resident Space Object (RSO) catalog using metric data obtained from RSO streaks gathered by astronomical telescopes. The focus of this work is on processing data from three possible streak categories: streaks that only enter, only exit, or cross completely through the astronomical image. Successful use of this data will aid in resolving uncorrelated tracks, space object identification, and threat detection. Incorporation of external data sources will also reduce the number of routine collects required by existing SSA sensors, freeing them up for more demanding tasks. The results clearly demonstrate that accurate orbital reconstruction can be performed using an RSO streak in a distorted image, without applying calibration frames and that partially bound streaks provide similar results to traditional data, with a mean degradation of 6.2% in right ascension and 42.69% in declination. The methodology developed can also be applied to dedicated SSA sensors to extract data from serendipitous streaks gathered while observing other RSOs.

*for my grandfather*

## **Acknowledgments**

The author would like to thank Air Force Research Laboratory (AFRL) for sponsoring this work and specifically Dr. Paul Schumacher of AFRL for inspiring the approach. Along with, Dr. Mark Spencer of AFRL, and Captain Tyler Hardy of Air Force Institute of Technology (AFIT) for their invaluable insights into computer modeling and simulation techniques. Mr. David Shultz, also of AFIT for assisting in the practical needs of gathering real world data and for providing alternative sources for astrometry methods. Captain Camdon Cady for the virtual machine and Linux tutoring that was required, and Captain Gary Goff for his Unscented Kalman Filter and orbital analysis tools. Finally, my entire committee, for believing in this crazy idea, and especially Dr. Jonathan Black and Dr. Richard Cobb for your many hours of mentoring and teaching.

On a personal note, I want to thank my wife and children, I know the past couple of years have not been easy and I could not have done it without your love and support. I would also like to thank my father for all those trips to his lab when I was young. There is a remarkable similarity between counting white dots under a microscope and counting white dots through a telescope. My grandfather to whom this work is dedicated for teaching me the value of hard work and always providing a positive example for how to live life. To the doctor who will “probably quit in the morning;” thanks for being there whenever I wanted to throw in the towel. Finally, my astrometry mentor and editor, you know who you are, for the many hours spent tirelessly working to make this document as good as it could be. This document would not have happened without all of your help.

Charlie T. Bellows

## Table of Contents

	Page
Abstract . . . . .	iv
Dedication . . . . .	v
Acknowledgments . . . . .	vi
Table of Contents . . . . .	vii
List of Figures . . . . .	xi
List of Tables . . . . .	xiv
List of Symbols . . . . .	xv
List of Source Code . . . . .	xviii
List of Acronyms . . . . .	xix
I. Introduction . . . . .	1
1.1 Research Motivation . . . . .	1
1.2 Research Objective . . . . .	3
1.2.1 Research Questions . . . . .	4
1.2.2 Assumptions and Limitations . . . . .	5
1.3 Dissertation Overview . . . . .	5
II. Background . . . . .	7
2.1 Space Situational Awareness Common Terms . . . . .	7
2.1.1 Space Situational Awareness Defined . . . . .	7
2.1.2 Astrodynamics . . . . .	8
2.1.3 J2000 Reference Epoch . . . . .	8
2.1.4 Earth Centered Inertial Coordinate System . . . . .	8
2.1.5 Uncorrelated Tracks . . . . .	9
2.1.6 Flexible Image Transport System . . . . .	9
2.2 Current Space Situational Awareness Sensors . . . . .	10
2.2.1 Optical Tracking Techniques . . . . .	10
2.2.1.1 Dual Angle Electro-Optical Tracking . . . . .	11



	Page
2.2.1.2 Electro-Optical Error Sources . . . . .	12
2.2.2 Tracking by Determining Most Probable Shape . . . . .	15
2.2.2.1 Why does Shape Matter? . . . . .	15
2.2.2.2 Attribute Estimation . . . . .	15
2.2.2.3 Methods to Determine Shape . . . . .	16
2.3 Current Space Situational Awareness Challenges . . . . .	18
2.3.1 Current Uncertainty in Low Earth Orbit Covariance . . . . .	19
2.3.2 Current Uncertainty in Geosynchronous Covariance . . . . .	19
2.3.3 High Area-to-Mass Ratio Objects . . . . .	20
2.3.4 Resolving Uncorrelated Tracks . . . . .	21
2.3.5 Data Association and Information Fusion . . . . .	21
2.3.5.1 Data Association: Single-Frame Methods . . . . .	23
2.3.5.2 Data Association: Multi-frame Methods . . . . .	23
2.4 Astrometry Techniques . . . . .	25
2.4.1 Affine Transform Astrometry . . . . .	26
2.4.1.1 Standard Coordinates . . . . .	27
2.4.1.2 Finding Plate Constants . . . . .	27
2.4.1.3 Extracting Position Coordinates from Streaks . . . . .	29
2.4.2 Frontal Pinhole Camera Model . . . . .	29
2.4.3 The Camera with Intrinsic Parameters . . . . .	32
2.4.4 <i>astrometry.net</i> . . . . .	33
2.5 Streak Identification and Isolation Research . . . . .	34
2.6 On-going Parallel Research . . . . .	35
2.6.1 Air Force Research Laboratory Research Efforts . . . . .	35
2.6.2 Search and Determine Integrated Environment . . . . .	36
2.6.3 Resolving Uncorrelated Tracks (UCTs) with Search and Determine Integrated Environment . . . . .	36
2.7 Orbit Determination . . . . .	37
2.8 External Sensor Data . . . . .	38
2.9 Summary . . . . .	40
III. Analysis of Angle Data Extraction Techniques . . . . .	41
3.1 Generating a Baseline Image . . . . .	42
3.2 Error Sources that Impact Position . . . . .	43
3.2.1 Tracking Error . . . . .	44
3.2.2 Modeling the Effects of a Lens with Aberrations . . . . .	44
3.2.3 Jitter . . . . .	46
3.2.4 The Atmosphere . . . . .	47
3.2.5 Multiple Error Sources . . . . .	49
3.3 Real-world Images . . . . .	49
3.4 Astrometry Methods Performance . . . . .	50

	Page
3.5 Baseline Image . . . . .	51
3.6 Tracking Error Impact . . . . .	53
3.7 Lens Aberration Impact Results . . . . .	54
3.8 Jitter Impact Results . . . . .	55
3.9 Atmospheric Turbulence Results . . . . .	55
3.10 Multiple Error Sources . . . . .	57
3.11 Real-World System Images . . . . .	57
3.12 Summary . . . . .	59
IV. Updating Track Data from Partial Satellite Streaks . . . . .	60
4.1 The Challenge . . . . .	60
4.2 Methodology . . . . .	62
4.3 Hardware Setup and Data Simulation . . . . .	62
4.3.1 Partially Bounded Streak Simulation . . . . .	62
4.3.2 Unbounded Streak Simulation . . . . .	63
4.4 Assumptions and Limitations . . . . .	64
4.5 Predicted Position . . . . .	64
4.6 Extracting Streak Data . . . . .	69
4.6.1 Streak Extraction . . . . .	69
4.6.2 <i>astrometry.net</i> . . . . .	70
4.7 Partial Streaks . . . . .	70
4.7.1 Determining Residuals . . . . .	71
4.8 Unbounded Streaks . . . . .	72
4.8.1 Determining Possible Position and Time . . . . .	73
4.9 Results and Analysis . . . . .	75
4.9.1 Partial Streaks Results and Analysis . . . . .	75
4.9.1.1 Position Accuracy when using Simplified Deep Space Perturbation Model (SDP4) . . . . .	75
4.9.1.2 Determined Residuals . . . . .	75
4.9.1.3 End Point Localization . . . . .	77
4.9.1.4 Impact to Current Operations . . . . .	80
4.9.2 Unbounded Streaks . . . . .	80
4.9.2.1 Position Accuracy . . . . .	81
4.9.2.2 Best Streak Observation . . . . .	82
4.9.2.3 Residuals Analysis . . . . .	83
4.9.2.4 Covariance Analysis . . . . .	85
4.9.2.5 Mahalanobis Distance Analysis . . . . .	88
4.10 Summary . . . . .	89

	Page
V. Conclusions and Future Work . . . . .	91
5.1 Phase I Conclusions . . . . .	91
5.2 Phase II Conclusions . . . . .	92
5.2.1 Using Partially Bounded Streaks . . . . .	93
5.2.2 Using Unbounded Streaks . . . . .	94
5.3 Future Work . . . . .	94
Appendix A: Image Processing Methods . . . . .	97
A.1 Data Reduction . . . . .	98
A.1.1 Detect Streak and Extract Pixel . . . . .	98
A.1.1.1 Detecting RSO streaks . . . . .	98
A.1.1.2 Determine Pixel Locations . . . . .	103
A.2 Convert Pixels to Spherical $\alpha$ and $\delta$ . . . . .	107
A.3 Right Ascension, Declination, and Time . . . . .	108
Appendix B: Verification and Validation of Angle Extraction Techniques . . . . .	109
Appendix C: Orbital Determination Unscented Kalman Filter . . . . .	127
Appendix D: Useful Tools . . . . .	132
Bibliography . . . . .	135

## List of Figures

Figure	Page
2.1 Typical data association conflict: prediction variance vs. observed objects . . .	24
2.2 Plate coordinates vs. standard coordinates . . . . .	28
2.3 Frontal pinhole imaging model . . . . .	30
2.4 Virtual focal length ( $f'$ ) as derived from field of view ( $\theta$ ) . . . . .	31
2.5 Transformation from pixel coordinates to $[XYZ]^T$ . . . . .	32
2.6 SADIE high-level architecture . . . . .	37
3.1 Simulated baseline image . . . . .	42
3.2 Aberrations effects: image . . . . .	46
3.3 Aberrations effects: pixels . . . . .	46
3.4 System jitter effects: image . . . . .	47
3.5 System jitter effects: pixels . . . . .	47
3.6 Atmospheric effects: image . . . . .	48
3.7 Atmospheric effects: pixels . . . . .	48
3.8 Multiple noise sources: image . . . . .	49
3.9 Multiple noise sources: pixels . . . . .	49
3.10 A complete RSO streak captured by TeleTrak (contrast enhanced) . . . . .	50
3.11 RMSE for each method and noise source . . . . .	51
3.12 RSO position offset introduced by method . . . . .	53
3.13 RSO position offset caused by tracking error . . . . .	54
3.14 RSO position offset introduced by aberrations . . . . .	55
3.15 System jitter RSO position offset . . . . .	56
3.16 Atmospheric turbulence RSO position offset . . . . .	56
3.17 RSO position offset introduced by multiple error sources . . . . .	57

Figure	Page
3.18 Real-world system image detected position . . . . .	58
4.1 A partial RSO streak captured by an optical sensor . . . . .	61
4.2 An RSO streak passing through a TeleTrak image . . . . .	61
4.3 A simulated partially bounded RSO streak . . . . .	63
4.4 A simulated unbounded RSO streak passing through the frame . . . . .	64
4.5 Camera model predicted star positions overlying true positions (px) . . . . .	68
4.6 The notional difference between a streak's centroid and detected end point . . .	71
4.7 A notional NET streak . . . . .	74
4.8 A notional NLT streak . . . . .	74
4.9 The determined right ascension residuals . . . . .	76
4.10 The determined declination residuals . . . . .	76
4.11 The right ascension difference between traditional and experimental methods .	77
4.12 The declination difference between traditional and experimental methods . . .	78
4.13 The difference in $x'$ pixels between edge and centroid location . . . . .	79
4.14 The difference in $y'$ pixels between edge and centroid location . . . . .	79
4.15 First pass through the optimizer . . . . .	82
4.16 $n^{th}$ pass through the optimizer . . . . .	83
4.17 Final pass through the optimizer . . . . .	83
4.18 Estimated $\alpha$ residuals of $2^{nd}$ and $3^{rd}$ night ( $1^{st}$ night unbound streak) . . . . .	84
4.19 Estimated $\delta$ residuals of $2^{nd}$ and $3^{rd}$ night ( $1^{st}$ night unbound streak) . . . . .	85
4.20 Estimated $\alpha$ residuals of $3^{rd}$ night ( $2^{nd}$ night unbound streak) . . . . .	85
4.21 Estimated $\delta$ residuals of $3^{rd}$ night ( $2^{nd}$ night unbound streak) . . . . .	86
4.22 Mean longitude covariance . . . . .	86
4.23 Semi-major axis covariance . . . . .	87
4.24 Mean longitude covariance . . . . .	87

Figure	Page
4.25 Semi-major axis covariance . . . . .	88
4.26 Night 1 Mahalanobis distance . . . . .	89
4.27 Night 2 Mahalanobis distance . . . . .	89
A.1 Image Processing Method . . . . .	97
A.2 Image ( $S_{xy}$ ) Returned by the System . . . . .	99
A.3 Mapped image background . . . . .	101
A.4 Image after mapped background removed . . . . .	102
A.5 The contrast enhanced image . . . . .	102
A.6 The star masked image . . . . .	103
A.7 The binary streak image . . . . .	104
A.8 The Hough Transform of the binary streak image . . . . .	106
A.9 The identified RSO streak . . . . .	107
B.1 Offset between systems . . . . .	110
B.2 Aligning the Zenith/Z axis . . . . .	110
B.3 Aligning East/X, South/Y . . . . .	111
B.4 The aligned system . . . . .	111
B.5 Offset between two systems . . . . .	111
B.6 Aligning the Zenith/Z axis . . . . .	111
B.7 Rotation about 3 <sup>rd</sup> Axis . . . . .	112
B.8 The aligned system . . . . .	112
B.9 Affine transform astrometry overlying true positions . . . . .	124
B.10 Pinhole camera model overlying true positions . . . . .	125
B.11 Intrinsic parameters calibrated model overlying true positions . . . . .	126

## List of Tables

Table	Page
3.1 Requirements for each astrometry method . . . . .	41
3.2 Root-mean-squared-error (RMSE) in degrees and kilometers . . . . .	52
4.1 Intrinsic and Extrinsic Parameters for the Optical System . . . . .	66
4.2 Predicted position RMSE for the Intelsat 1R TLE . . . . .	69
4.3 Simulated Data Sets from Collected Imagery of Intelsat 1R . . . . .	72
4.4 RMSE for positional data . . . . .	78

## List of Symbols

$\alpha$	= right ascension (degrees)
$\alpha_0$	= Earth Centered Inertial (ECI) right ascension to the principal point of the image referenced to a J2000 epoch
$\beta$	= azimuth (degrees/radians)
$\beta_0$	= Azimuth of the bore sight
$\nu$	= the difference between the observed and the calculated observation (residuals)
$\delta$	= declination (degrees)
$\delta_0$	= ECI declination to the principal point of the image referenced to a J2000 epoch
$\eta$	= aspect ratio
$\hat{\mathbf{P}}$	= updated covariance matrix
$\hat{\mathbf{x}}$	= updated state estimate
$\hat{\mathbf{y}}$	= an anticipated observation vector
$\lambda$	= arbitrary scale factor
$\rho_{\mathbf{IJK}}$	= An ECI position vector with a J2000 epoch
$\mathbf{P}$	= the state covariance matrix
$\mathbf{Q}$	= the process noise covariance
$\mathbf{R}_{\text{noise}}$	= the observation noise covariance matrix



**R** = Rotation matrix

**S** = the predicted observation covariance

**t** = translation vector

$\mu_{\oplus}$  = gravitational parameter of Earth ( $km^3/(s^2)$ )

$\bar{\mathbf{P}}$  = the propagated covariance

$\Psi$  = the Mahalanobis distance

$\rho$  = position vector in the South, East, Zenith reference frame

$\rho_x, \rho_y, \rho_z$  =  $1^{st}$ ,  $2^{nd}$ , and  $3^{rd}$  vector components of  $\rho$

$\sigma$  = the standard deviation of the error

$\theta$  = the angle defined by the motion of travel of the RSO (radians)

$\theta_{asec}$  = the angle define the motion of travel of the RSO (arc-seconds)

$a - f$  = plate constants

$d$  = distance traveled by RSO during  $t_{exposure}$ (km)

$d_{px}$  = the distance the RSO traveled in pixels

$el$  = elevation (degrees/radians)

$el_0$  =Elevation of the bore sight

$f'$  = systems virtual focal length (px)

$k$  = system point spread function

$n$  = number of state elements

$n_0$  = random additive noise

$O_x, O_y$  = pixel coordinate origins (px)

$pixscale$  = the arc-seconds contained in a single pixel

$r$  = orbital radius ( $km$ )

$r_{\oplus}$  = radius of the Earth

$s_x, s_y$  = Scale factors (radians/px)

$s_{\epsilon}$  = skew factor

$t$  = exposure time of the image (s)

$T_*$  = Translation of coordinate \*

$v$  = velocity of the RSO ( $km/s$ )

$w$  = covariance weights

$X, Y, Z$  = real world coordinates

$x', y'$  = image coordinates

$x_s, y_s$  = Pixel coordinates (px)

## List of Source Code

A.1	Script used to reduce images to pixel coordinates . . . . .	98
A.2	Process an image to identify RSO streaks . . . . .	99
A.3	Finding the pixel location of image streaks . . . . .	104
B.1	Affine transform model source code . . . . .	114
B.2	Pinhole camera model source code . . . . .	115
B.3	Camera calibrated with intrinsic parameter Model Source Code . . . . .	117
B.4	Performing an affine transform . . . . .	119
B.5	Determining the pinhole camera model vector . . . . .	120
B.6	Rotating the camera model vector to South, East, Zenith (SEZ) . . . . .	121
B.7	Transforming SEZ to look angle . . . . .	122
B.8	Determining the calibrated camera vector . . . . .	122
B.9	Rotating the calibrated camera vector to SEZ . . . . .	123

## **List of Acronyms**

Acronym	Definition
AF	Air Force
AFIT	Air Force Institute of Technology
AFRL	Air Force Research Laboratory
AFSPC	Air Force Space Command
AGI	Analytical Graphics Inc.
API	Application Programming Interface
ASAT	anti-satellite
CCD	Charge-Coupled Device
COTS	Commercial Off the Shelf
DARPA	Defense Advance Research Projects Agency
DoD	Department of Defense
ds9	Smithsonian Astrophysical ObservatoryImage ds9
ECI	Earth Centered Inertial
EO	Electro-Optical
fits	Flexible Image Transport System
FOV	Field-of-View
GEO	Geosynchronous Orbit
GEODSS	Ground Based Electro-Optical Deep Space Surveillance
GNN	Global Nearest Neighbor
GUI	Graphical User Interface
HAMR	High Area-to-Mass Ratio
imwcs	Image World Coordinate System Setting Utility
IPAC	Infrared Processing and Analysis Center

Acronym	Definition
IRAF	Image Reduction and Analysis Facility
JMS	Joint Space Operations Center Mission System
jpg	Joint Photographic Expert Group
JSpOC	Joint Space Operations Center
LADAR	Laser Radar
LEEDER	Laser Environmental Effects Definition and Reference
LEO	Low Earth Orbit
LSST	Large Synoptic Survey Telescope
LST	Local Sidereal Time
MHF	Multi-Hypothesis Filter
MHT	Multiple-Hypothesis Testing
MMAE	Multiple-Model Adaptive Estimation
NASA	National Aeronautics and Space Administration
NITARP	National Aeronautics and Space Administration Infrared Processing and Analysis CenterTeacher Archive Research Program
NN	Nearest Neighbor
NOMAD	Naval Observatory Merged Astrometric Dataset
NORAD	North American Aerospace Defense Command
NRC	National Research Council
NRL	Naval Research Laboratory
Pan-STARRS	Panoramic Survey Telescope and Rapid Response System
PC	Personal Computer
PSF	point spread function
RMSE	root-mean-squared-error
ROTAS	Report Observation Association
RSO	Resident Space Object

Acronym	Definition
SAD	Search and Determine
SADIE	Search and Determine Integrated Environment
SAO	Smithsonian Astrophysical Observatory
SID	Satellite Identification
SEZ	South East Zenith
SOI	Space Object Identification
SP	Special Perturbations
SRP	Solar Radiation Pressure
SSA	Space Situational Awareness
SSN	Space Surveillance Network
SSS	Space Surveillance System
STK	Systems Tool Kit
TEA	Time, Elevation, and Azimuth
TLE	Two Line Element Set
UCAC	U. S. Naval Observatory Charge-Coupled Device Astrograph Catalog
UCT	Uncorrelated Track
USNO	U. S. Naval Observatory
WCS	World Coordinate System

# LEVERAGING EXTERNAL SENSOR DATA FOR ENHANCED SPACE SITUATIONAL AWARENESS

## 1. Introduction

### 1.1 Research Motivation

Recent space activities, including the 2007 anti-satellite (ASAT) test by China and the 2009 collision between a non-operational Kosmos satellite and an operational Iridium satellite, have increased global concern for reliable Space Situational Awareness (SSA) [1, 2]. The current U. S. National Space Policy requires timely and accurate global SSA [3]. According to this policy document, the SSA community should utilize information from commercial, civil, national security sources, and foreign partners to detect, identify, and attribute actions in space that violate long-term sustainability of the space environment. The policy also calls for the development and modernization of SSA capabilities [3].

In 2013, General Shelton, commander Air Force Space Command (AFSPC), stated that SSA provides the Air Force the ability to see and understand threats on Earth and in space, enabling successful operations in multiple domains [4]. The Joint Space Operations Center (JSpOC) has stated that one of their limiting factors for the desired level of SSA is a lack of available sensors [5]. Although all orbit regimes are a concern, the Air Force Research Laboratory (AFRL) is currently researching methods to characterize the Geosynchronous (GEO) regime. GEO challenges include dim and infrequently observed objects, making orbital determination difficult and sustainable SSA elusive [6].

Recent efforts by AFRL have focused on developing an automated system known as Search and Determine Integrated Environment (SADIE). SADIE is a Multi-Hypothesis Filter (MHF) designed to automate the current time-intensive process of resolving

Uncorrelated Tracks (UCTs) and updating the space catalog [7]. UCTs are defined as observed tracks (or objects in space moving across the Field-of-View (FOV) of a sensor) that do not correspond to known objects. SADIE performs its mission by inputting observational data: Time, Elevation, and Azimuth (TEA) and range, from both Electro-Optical and radar sensors around the world and using MHF to resolve these tracks by matching them to objects whose position is currently unknown. Unfortunately SADIE is limited by the data provided, requiring a minimum of four observations for UCT resolution, with more observations providing a higher confidence in the result [7]. Also of note is that angles-only solutions, provided by Electro-Optical (EO) sensors and including only right ascension, declination, and time, are more accurate when geographically separated sites are used to gather the observations required [8, 9]. Again, this geographic separation is an area with which AFSPC struggles, as there are only three optical sites used by the Space Surveillance Network (SSN). Located on Diego Garcia, at Socorro, New Mexico, and Maui, Hawaii these sites provide limited views of the sky and are tasked with all AFSPC required optical observations [10].

It is apparent that a need for more SSA sensors exists, but these sensors don't need to be dedicated to the SSA mission. Currently the SSN uses several contributing sensors to provide additional data sources as required, and the Defense Advance Research Projects Agency (DARPA) is also exploring the idea of using sensors part time as part of the Orbital Outlook program. This outsourcing demonstrates that EO sensors designed for other missions can also be tasked to collect SSA data. This work herein will explore the idea of using non-dedicated SSA sensors as an additional data source for enhancing SSA.

Many sensor systems, such as EO systems used by astronomers, routinely record satellites while performing their intended scientific purpose. This work explores the idea of using these serendipitous collects to improve the global SSA environment. Unlike contributing sensors, or those used by the Orbital Outlook program, the sensors considered



in this work referred to simply as ‘external sensors,’ will be performing a mission other than SSA and providing any images including conveniently captured satellites for analysis. To achieve enhanced SSA it is desirable for these sensors to be able to detect dim objects from geographically separated locations. These sensors should provide the same TEA data as current dedicated SSA sensors.

## **1.2 Research Objective**

Several potential SSA resources exist outside the Air Force. Currently it is unknown if data from these resources can be included in the global SSA environment. This inclusion needs to be accomplished using a method that ensures the data is trustworthy, usable, and will not corrupt the data that has already been validated by the SSN.

One potential resource is the astronomy community, as the serendipitous streaks collected during astronomical observations may be very beneficial to the SSA mission. Furthermore, other telescope networks that observe man-made space objects often referred to as a Resident Space Object (RSO), such as the Exoanalytic<sup>1</sup> network, Orbital Outlook, or Air Force Institute of Technology (AFIT)’s TeleTrak system, can also provide useful SSA data [11–13]. Any increase in available resources will also relieve the demands being placed on the SSN providing not only more data, but also allowing the Ground Based Electro-Optical Deep Space Surveillance (GEODSS) sensors to focus on more demanding tasks, such as UCT resolution, Space Object Identification (SOI), or threat detection.

The objective of this research is to determine requirements for external sensor data to be of value for SSA. This work focuses on addressing the concerns of the SSA community for using sensors not controlled by the Department of Defense (DoD) by seeking to determine a data calibration method that allows for reliable data extraction. Methods for using the data provided by the network will be explored; the ability to perform orbital

---

<sup>1</sup>Exoanalytic is a company offering technical solutions to government and private industry in the areas of space, missile defense, and research & development; Mission Viejo CA

determination seems obvious, but the data requirements for this application are strict. Every observation collected will contain both along-track and cross-track data. Along-track error is the largest error source in the current Two Line Element set (TLE); documented in Section 2.3. This research will seek to experimentally demonstrate the extraction of right ascension, declination, and time data from distorted images and provide it for enhanced SSA.

### *1.2.1 Research Questions*

This research will be broken into two phases, each phase addressing necessary questions to determine the efficiency of external sensors for SSA. The first phase, presented in Chapter 3, will seek to find the best data calibration technique for extracting angle data from images. This objective will be accomplished by comparing several existing techniques to determine the sensitivity of each process to the distortions found in EO data. The focus of Phase I work will be to answer the following research questions:

1. What is the level of accuracy achieved by each calibration method?
2. Which method best accounts for image distortion?

The work of Phase II, presented in Chapter 4, examines the feasibility and military utility of performing in-track and cross-track updates for an RSO catalog using metric data obtained from streaks gathered by astronomical telescopes. Traditional orbital determination uses bounded streaks, along with image time and exposure duration, to determine an accurate observation time and develop position vectors. Identification of the center of a streak is used to provide precise timing of the RSO [8, 14]. The focus of this work is on gathering and using three possible data categories: streaks that only enter, only exit, or cross completely through the astronomical image. Streaks that only enter or only exit an image have incomplete timing data and using them relies on proper end point detection. While streaks that pass completely through the image provide no time

data, requiring an accurate time estimate before they can be used. The focus of Phase II work will be to answer the following research questions:

1. Can observations of tracks only partially in the image be used to improve SSA?
2. Can observations of tracks passing fully through the image be used to improve SSA?

### ***1.2.2 Assumptions and Limitations***

This research assumes that the stars in Naval Observatory Merged Astrometric Dataset (NOMAD) catalog are a representative subset of the U. S. Naval Observatory Charge-Coupled Device Astrograph Catalog (UCAC) used by DoD. Both catalogs are developed and maintained by the U. S. Naval Observatory (USNO), but as the NOMAD catalog is much easier to obtain than the UCAC, NOMAD will be used as the star catalog for this research.

The work will be limited to observing GEO satellites with TLE available in the public catalog. Data will further be limited to what can be collected by the TeleTrak system at AFIT. Effort will be taken to make these images as representative of larger systems as possible but the FOV observed will be 2-4x smaller than what would be expected from a larger survey telescope used by astronomers. Finally, this research will focus on using images provided by telescopes in sidereal tracking mode.

## **1.3 Dissertation Overview**

This document is divided into five chapters and four appendices. Chapter 2 contains a literature review of topics relevant to the research that follows, divided into eight sections. Section 2.1 defines basic terminology used in subjects relevant to the research. Section 2.2 covers current SSA methods. Section 2.3 discusses various challenges faced by the community. Section 2.4 covers alternative methods for determining position vectors used in astrometry. Sections 2.5 and 2.6 provide an overview of influential and parallel research that both benefit and enable aspects of this work, further demonstrating that the focus of this

research should be on the validation of the data produced by the sensors. Section 2.7 covers the basics of orbital determination. Section 2.8 discusses the usefulness of astronomy data and missions.

Chapter 3 provides the method and analysis used to determine the best astrometry approach for processing Electro-Optical data in the experiments that follow. Chapter 4 covers the experiments used to determine the usefulness of partial and full streaks in imagery and provides the analysis of the experimental results. Chapter 5 reports the findings of this research and draws conclusions based on the analysis and suggested future work.

The document concludes with four appendices. Appendix A sets up the image processing and calibration process and its verification. Appendix B provides the validation and verification of the astrometry models tested in Chapter 3. Appendix C discusses the use of Goff's algorithms to perform Unscented Kalman Filter (UKF) orbital determination used to determine the usefulness of full RSO streaks [15, 16]. Finally, Appendix D covers current options for tools that can be used to calibrate the images, extract orbital elements, and return meaningful results.

## **2. Background**

This chapter is designed to motivate the research, define terms, and establish the context of similar scholarly work. The chapter begins by providing important definitions within the SSA community. An exploration of current SSA techniques will then be provided followed by a discussion of work conducted in the past as well as ongoing work related to this effort. Finally, the chapter will explore alternative methods for conducting some of the critical aspects in SSA that exist outside the community but may be useful for image calibration.

### **2.1 Space Situational Awareness Common Terms**

#### ***2.1.1 Space Situational Awareness Defined***

According to the Space Foundation, SSA refers to the ability to “view, understand and predict the physical location of natural and man made objects in orbit around the Earth” [17]. AFSPC SSN Site Information Handbook defines SSA as “the battle-space awareness required for planning, executing, and assessing protection of space assets, prevention of hostile actions, and negation of hostile resources in all mediums.” The handbook goes on to state that SSA is a result of having sufficient knowledge about current and planned “space-related conditions, constraints, capabilities and activities” [10]. The National Space Policy states that SSA is created using “space surveillance, intelligence and other information” [3]. Recent activities on orbit, including the 2007 ASAT test by China and the 2009 collision between a non-operational Kosmos satellite and an operational Iridium satellite, have increased global concern for reliable SSA. Space surveillance includes various methods of detection and tracking of RSOs using astrodynamics techniques developed in the last 50 years.

### ***2.1.2 Astrodynamics***

The field of astrodynamics seeks to apply the laws of celestial mechanics to the orbits of man-made objects around the Earth or other celestial bodies. These orbits are fundamentally derived using Newtonian mechanics but also must account for perturbing forces acting upon these bodies. Perturbing forces include atmospheric drag, solar radiation pressure, and gravitational forces that require more detailed analysis than needed for celestial objects [6].

### ***2.1.3 J2000 Reference Epoch***

In both celestial and orbital mechanics, an epoch is a reference point in time for some time-varying quantity, including orbital elements and celestial coordinates as they are subject to time varying perturbations [18]. The primary function of an epoch is to specify the parameters of motion. An epoch is always referenced to a set time such as the Julian epoch (J2000) the standard epoch in use today by astronomers. The Julian epoch is exactly 1200 terrestrial time on January 1, 2000 in the Gregorian calendar [19]. This epoch will be used in Chapters 3 and 4 to define the motion of both the stars and satellites.

### ***2.1.4 Earth Centered Inertial Coordinate System***

Also important in defining the location of RSOs is the reference system used. The preferred reference system for reporting satellite locations to the JSpOC is the Mean Equator, Mean Equinox (MEME) ECI system. The ECI frame is defined using the Earth's orbital plane and the orientation of the Earth's axis of rotation [8], where the first axis points in the direction of the vernal equinox, the second axis is 90° east along the ecliptic plane, and the third axis is along the Earth's axis of rotation.

In this coordinate system the position of both satellites and stars are given in right ascension ( $\alpha$ ), and declination ( $\delta$ ). Right ascension is defined to be measured positive to the east in the plane of the equator from the direction of the vernal equinox; although it can

be reported from  $0^h - 24^h$ , in this work degrees,  $0^\circ - 360^\circ$ , are used. The declination is measured northward from the equator ( $0^\circ - 90^\circ$ ) [8].

Throughout this work unless otherwise noted, satellite positions will be given in ECI right ascension and declination. Other coordinate systems used will be discussed as they appear throughout the work along with the required transformation to the ECI coordinate system.

### ***2.1.5 Uncorrelated Tracks***

The term Uncorrelated Track (UCT) is a reference to tracked objects that do not correlate with known catalog objects. They are most commonly created through on-orbit breakups, maneuvering satellites, and small objects that are only occasionally tracked. Each day numerous UCTs occur and performing correlation of these objects to create the required ephemeris data for catalog maintenance is currently a manual process [20]. Recent efforts by Air Force Research Laboratory (AFRL) are attempting to automate this process using a program called Search and Determine Integrated Environment (SADIE), see Section 2.6.2. For more on UCTs as described in Section 2.3.4.

### ***2.1.6 Flexible Image Transport System***

The most common type of file produced by both observational astronomy and non-SSN EO sensors is a Flexible Image Transport System (fits) file-format. The fits file is composed of a plain text header followed by a binary image. The header at a minimum includes information coordinates of the image source location in World Coordinate System (WCS). The header may also contain: target, telescope, astronomer, date observed, wavelength used, and any data reduction steps taken. The presence of this data depends on the equipment used to collect the image and the settings input by the observer. The binary image can be composed of a single image plane, or many planes of images. The ‘image’ can also be a spectrum or stack of spectra, or a table of data [21].

A fits image is incredibly data rich, consisting of 16 to 32 bits per pixel. In other words there are at least  $2^{16}$  (65,536), and possibly as many as  $2^{32}$  (4.3 billion), discrete levels of data for each pixel. Because of the level of information contained in each pixel, it is worth noting that saving a fits as another file type results in the loss of considerable data. For example a Joint Photographic Expert Group (jpg) file format only stores  $2^8$  (256) discrete levels of data. Therefore saving a fits file as a jpg results in the compression of data from 65,536 levels to merely 256 levels [22]. This work will focus on using fits images provided by astronomical sensors to prevent the data loss that occurs when changing file types.

## **2.2 Current Space Situational Awareness Sensors**

Currently AFSPC uses the SSN as a primary means of achieving SSA. The SSN is comprised of phased array radars, mechanical radars, optical systems and a space-based optical system. In the past the SSN also had a multistatic radar, but that was shut down on 1 October 2013 [23]. These systems are covered in detail in the AFSPC SSN Site Information Handbook and will only be discussed in this document as required for understanding in the context of the research[10]. Other countries also have systems that contribute to global SSA. Weeden et al. provides an excellent survey of these systems that can be found in [24].

### ***2.2.1 Optical Tracking Techniques***

Optical tracking can be broken into two methods: Sidereal track and Rate track. The Sidereal track method employed by telescopes is accomplished when the telescopes FOV moves across the sky at the same rate as the stars appear to move. This tracking method keeps the star field static in the image taken; however any satellites in the image will appear as streaks of light across the image. When a GEODSS telescope employs this method it



can use the fixed star image for photometric<sup>2</sup> calibration, as described in Section 2.2.2.3 [10].

Rate track methods are performed by having the telescope move to keep the satellite stationary in the image. When this method is utilized, the satellite appears as a single white dot and the stars appear as streaks in the image plane. Both the sidereal and rate track methods are employed by GEODSS to image satellites to collect data[10].

Traditional orbital determination techniques use streaks bounded within the image, along with image time and exposure duration, to determine an accurate observation time and corresponding position vectors. Identification of the midpoint of a streak is used to provide precise timing of the RSO as it correlates to the middle of the exposure which can be determined with high accuracy [8, 14, 25].

GEODSS are dedicated SSN sensors tasked with providing timely astrometric and photometric data on tasked deep space<sup>3</sup> objects in support of the Space Surveillance mission. Currently there are three operational GEODSS sites located in Socorro, NM; on the island of Diego Garcia in the British Indian Ocean Territories; and finally on the island of Maui, HI [10]. This distribution of telescopes provides complete coverage of the GEO belt [24]. Each of these sites has three telescopes with a 40-inch aperture and a two-degree FOV. The optical system is capable of detecting objects 10,000 times dimmer than the human eye and is naturally limited by local weather conditions [10]. These telescopes provide the majority of all deep space and GEO tracks per day for the JSpOC SSA Cell [10].

#### ***2.2.1.1 Dual Angle Electro-Optical Tracking***

Both Gauss and Laplace developed methods for determining the orbit of distant objects based solely on visual observations, a serious challenge in mathematical astronomy

---

<sup>2</sup>The goal of photometric calibration is to accurately measure the intensity (per wavelength) of the light captured

<sup>3</sup>RSOs are considered to be in deep space if there orbital period is greater than 225 *minutes* [26]

at the time. Today these methods are widely used to determine the orbits of satellites. Dr. Pedro Ramon Escobal provides detailed solutions to both methods in his book *Methods of Orbit Determination* [14].

The basic concept of the Gaussian method is to take a minimum of three line of sight vectors and their corresponding times and convert them to the three position vectors required to “fit” an orbit. If more than three vectors are available, the solution can be improved using a number of algorithms. For the conversion to be accurate, several site characteristics must be known including the observer’s Local Sidereal Time (LST), latitude and longitude, and altitude as well as the target azimuth and elevation. For ease of calculations all measurements are converted to the ECI coordinate system.

Unfortunately, no system is perfect and the data of dual angle observations will not be without error. Therefore it is necessary to note the level of precision and accuracy provided by the equipment used for these calculations. Typically, as pointed out by Wiesel, systems are designed by finding and eliminating all large error sources until the cost of doing so exceeds the benefit of increased accuracy [9]. The cost to accuracy ratio is a design parameter driven by requirements; therefore, different telescopes will have varying accuracy that needs to be accounted for when using an optical system for SSA.

#### ***2.2.1.2 Electro-Optical Error Sources***

A Strehl ratio is defined as a measure of the quality of optical image formation. This concept was originally proposed by Karl Strehl and bears his name. The Strehl ratio is used in situations where optical resolution has been compromised by any error sources. The ratio always has a value between zero and one with a perfect optical system receiving a one [27]. Error sources for optical images include atmospheric distortions, instrument distortions, and sensor distortions. Each of these areas are discussed below.

The Earth’s atmosphere is the largest contributor to distortion of any system looking into space from the Earth’s surface. Some clouds completely block the sensor’s FOV while

others may simply “dim” the light passing through them. Furthermore water vapor and dust also affect light in time varying ways while still allowing light to reach the sensor. Finally higher-order atmospheric effects such as scintillation, moment to moment variations in the atmospheric refraction, and optical density changes in airflow past the aperture also impact the observed point spread function [27].

The atmosphere can be accounted for using an atmospheric model, such as AFIT’s Laser Environmental Effects Definition and Reference (LEEDER) program [28]. Alternatively if the data is available, the current atmospheric effects can be determined using the stars available in the image in the same way that Nishimoto did in the late 1990’s [29]. Another option is to electronically shift images from one Charge-Coupled Device (CCD) cell to another to counteract atmospheric blur, such a system is currently used by Panoramic Survey Telescope and Rapid Response System (Pan-STARRS) [30]. To illustrate the effect on possible astrometry methods, the impact of atmospheric turbulence on an optical system is modeled in Section 3.2.4.

Sources of instrument distortion include jitter, obscuration Strehl, and higher-order aberrations: defects in optics manufacturing, and the heating/cooling of optical elements [27]. Typically an optical system will include measures to minimize some of these effects. Other effects are often accounted for by post-processing the data. The three standard levels of correction include raw uncorrected data (Level 0), optical and sensor corrected (Level 1A/B), and ortho-rectified corrected where straight lines should be straight and a given separation angle between two stars is matched in the image [31]. To aid in the determination of the best astrometry method for this work the impact of lens aberrations and jitter are modeled in Sections 3.2.2 and 3.2.3,.

Optical sensors used for both SSA and astronomy are designed and maintained to minimize the impact of the above distortion effects. Known as diffraction-limited systems

these sensors have the ability to produce images with angular resolution as good as the instrument's theoretical limit [32].

CCD technology is the preferred data generation method in both astronomy and SSA due to the CCD's sensitivity and noise characteristics. Unfortunately these devices are not perfectly uniform; quantum efficiency varies slightly from pixel to pixel, dark current is generated, and only a portion of each pixel is actually photo-reactive.

Photons arrive randomly, during any particular sub interval, therefore the photons that arrive during an exposure may be greater or less than the mean rate. This phenomenon is known as Poisson Noise. When dealing with faint objects, enough photons need to be collected to make a good image, thus some finite integration time is required. Long integration times can obscure real variations in the flux of photons from a source impacting the light curve <sup>4</sup> data [33].

There are also differences on the order of 1% from one photo-cell to the next between number of incoming photons and the number of electrons liberated. This quantum efficiency effect, known as non-uniformity, occurs as the photon signal is converted into electrons and needs to be accounted for [33]. Another source of error is the creation of free electrons by a CCD (dark current), and varies with temperature roughly doubling every 6° Celsius due to thermal agitation [33].

CCDs are adjusted to give an output slightly above zero when no signal is detected. Therefore each CCD has a zero point bias. Each pixel in the image contains a measurement of the system electrical offset, which is called the signal bias [33].

Quantization arises from digitizing the analog signal. This effect imposes a “granulation” on how accurately we know the signal coming out of the CCD camera [33].

The attachment of a CCD to the optical system also creates concerns. Vignetting occurs because the optical system does not illuminate the outer edges of the CCD as fully

---

<sup>4</sup>Photometric measurements of an RSO are commonly referred to as the 'light curve'

as it does the center. Finally dust shadowing caused by bits of dust on a filter or window that cast shadows on the sensor affect the image [33].

Methods to calibrate sensor output exist for all of these sensor effects, commonly referred to as dark fields, bias frames, and flat field frames. When properly collected, these frames correct the variations (non-uniformity) from the sensor, correct for the bias in the sensor and correct for any variations in illumination created by the optical system [33].

### ***2.2.2 Tracking by Determining Most Probable Shape***

Another application of EO to the field of orbital mechanics is to fuse shape information with orbital information to create a more accurate future position prediction. Crassidis has done considerable work in this field and his techniques are discussed below.

#### ***2.2.2.1 Why does Shape Matter?***

Knowledge of a space object attributes, such as mass and shape, enables more accurate non-conservative force models. RSO attributes can also help uniquely identifying UCT or assist in discriminating between two space objects on similar tracks. Identification of properly cataloged space objects is crucial to accurately predict future behavior, preventing further UCTs, and is critical for predicting conjunctions [34].

At GEO the largest non-conservative acceleration force is caused by Solar Radiation Pressure (SRP). This force is not guaranteed to be directly away from the sun as size, shape, attitude, and material properties all affect how satellites are affected by SRP. Unfortunately these attributes may all be currently unknown, and may change as they age and re-orient requiring them to be determined as a function of time [34].

#### ***2.2.2.2 Attribute Estimation***

According to Crassidis the objective of optical SOI is to “identify, attribute, and track characteristics of space objects from photometric data.” This method allows for the characterization of RSO to determine the properties of the: shape, surface, and material

along with dynamic states. Dynamic states can be used to classify RSOs further as either uncontrolled (debris or rocket body) or controlled (active satellite) [34].

### ***2.2.2.3 Methods to Determine Shape***

Numerous estimators exist for estimating the shape of an observed object. These methods include radar cross-sectioning approaches, as discussed by Toru Sato et al. in [35], and range Doppler interferometry, which is discussed in detail in [36]. In Low Earth Orbit (LEO) some objects can even be imaged by ground based telescopes. Often these methods are limited by the size and distance of the RSO to the observer [37].

Laser Radar (LADAR), which returns a three dimensional scan of an object, has also successfully been used to return the shape of an object. An advantage of using LADAR is the detail of the shape returned. Unfortunately the range of LADAR limits its use to proximity or rendezvous operations [38].

Lichter and Dubowsky cover an application for LADAR proximity operations for shape matching that is of interest here both for its use of kinematic data fusion and because it determines orbital characteristics as part of the solution method. Lichter and Dubowsky take range image data from a synchronized team of LADAR or stereo cameras and reduce it into coarse kinematic information at each sample time. This data is then passed to a Kalman filter, which forms the core of the estimation architecture [39].

The Kalman filter extracts the full dynamic state and inertial parameters of the object using the surrogate measurements, along with an accurate dynamic model. The Kalman filter provide estimates of both rotational and translational positions and velocities of the RSO. Once accurate information about the trajectory of the target is known, shape estimation reduces to a classic stochastic map-building problem [39].

As the target motion has been determined with respect to the sensors, pixel-level data can be fused into a probabilistic map of the target's shape using sensor noise models. This

method can be used for small satellites, debris, and large flexible space structures, but as mentioned above, the LADAR sensor has a significantly limited range [39].

An alternative method to LADAR which uses natural illumination is to observe and measure the reflected sunlight from an RSO. Sunlight reflected off of a satellite is fundamentally different than direct sunlight. The fundamental difference in the wavelength of reflected sunlight can be attributed to the reflectivity of the surface materials, the object's size and orientation, and the medium the light travels through along the line of sight to the observer. This fact can be leveraged to determine one or more of these properties from the light curve of an observed object [38]. In fact, astronomers have been using light curve data to study asteroid characteristics for some time [40].

It is crucial when working with light curve data that the observer accounts for environmental factors that impact the light curve. Most notably for RSOs, the atmosphere must be accounted for. In a study performed by Nishimoto et al. from 1999 to 2001, a method was developed to account for the atmospheric impact on light curve data. Background measurements of well understood visible stars were taken each night of observations; these observations allowed the system to account for atmospheric and instrument affects on the reflected light [29]. Once the external affects were accounted for and removed the wavelength of the reflected light was known. The reflected light can be compared to the absorptivity of known satellite materials to determine the material surface that reflected the sunlight as was demonstrated by Cauquy et al. in 2006. These measurements can also be used to identify two closely spaced spacecraft built out of different material (aluminum and gold solar blankets for example) [41].

Linares et al. presented a method in 2013 using an Multiple-Model Adaptive Estimation (MMAE) approach to identify the most probable shape of an orbiting object by comparing it to a catalog of candidate shape models. MMAE uses a set of parallel filters, each hypothesizing its own shape, to determine the probable shape of the object in

question. The filters used compared a known shape to light curve data collected by an EO system, the details of which are given in Reference 38.

The coupling between RSO attitude and non-conservative accelerations, SRP and atmospheric drag, can be exploited to aid in the estimation of an object's trajectory. SRP acceleration is a function of the same properties provided by light curve methods, such as the one developed by Linares et al. Therefore these methods also allow for the determination of the object's inertial orientation and trajectory. While the parallel filters are checking the shape they also simultaneously reduce the passively collected EO data to infer the unknown state vectors composed of the object's position, inertial-to-body orientation, and respective temporal rates [38]. The Linares et al. method is preferred to those mentioned above because it can be used on dim objects in GEO, relying only on light curve data.

As covered above several optical techniques are currently in use in the SSA field. Ground and space based sensors are used to track satellites both through angles only techniques and shape estimation. These techniques also face challenges ranging from the accuracy of orbital positions to infrequently observed dim objects.

### **2.3 Current Space Situational Awareness Challenges**

The National Research Council (NRC) recently found that current software suites used to perform astrodynamics calculations cannot meet current requirements to perform critical tasks such as conjunction analysis, and furthermore, they struggle with even fundamental tasks such as statistical orbit determination. The current system also makes it difficult to incorporate changes, including those that would enhance the system. The system in use by JSpOC is also not flexible enough to meet all user needs [6]. This section summarizes the findings of the NRC in their report titled 'Continuing Kepler's Quest-Assessing Air Force Space Command's Astrodynamics Standards,' the full report can be found in Reference 6.



As future sensors are able to resolve smaller objects the problem of correlating each new object will increase leading to a growing number of UCTs. Many commercial companies have demonstrated innovative algorithms that address specific issues faced by the SSA community. Unfortunately none of the alternatives addresses the full requirement set [6].

### ***2.3.1 Current Uncertainty in Low Earth Orbit Covariance***

According to Osweiler, using just the publicly available TLE generated by the current SSN, LEO satellites have their largest prediction error in the in-track velocity component of the satellite position vector. This error ranges from  $130 \text{ km}^2$  to  $4600 \text{ km}^2$ . Osweiler attributes this error, both magnitude and direction, to atmospheric drag [42].

### ***2.3.2 Current Uncertainty in Geosynchronous Covariance***

Accuracy of TLE data for Geostationary orbits and specifically Intelsat satellites have been studied in great detail. Chan and Navarro compared the publicly accessible TLE with INTELSAT proprietary orbital elements determined using their own dual station ranging system. The purpose of their comparison was to find out how reliable the TLE was for conjunction detection analysis. After comparing data for a full year Chan found that the orbital differences showed an average of 30-60 km difference in position, when the most recent TLE was propagated to the observational epoch [43]. Further studies have been performed on the accuracies of GEO TLEs in general. Früh and Schildknecht performed a four year study and after evaluating 13 GEO objects found an average of  $25 \text{ km}$  along-track and  $10 \text{ km}$  cross-track difference after propagating the most current TLE to the current epoch. They also found that the error grows the further from the TLE epoch, reaching hundreds of kilometers of track error in just a couple of days [44].

Osweiler's study shows that similar to LEO satellites, GEO satellites' largest observed prediction error from the TLE is also in-track or along the velocity component. compared to LEO satellites, Osweiler found that GEO satellites have a much smaller covariance matrix

overall. The magnitude of error varies by orbit as well. Satellites in stable orbits with negligible external forces have an average covariance uncertainty of approximately  $1 \text{ km}^2$ . While satellites affected by gravity variations or SRP have a covariance range from less than  $1 \text{ km}^2$  to greater than  $25 \text{ km}^2$  [42]. Recent work performed by Sabol et al. for AFRL found similar results in covariance uncertainty [7, 45].

### ***2.3.3 High Area-to-Mass Ratio Objects***

An important category of GEO objects is the High Area-to-Mass Ratio (HAMR) object. HAMR objects tend to have area-to-mass ratios between 1 and  $40 \text{ m}^2/\text{kg}$  [46]. Schildknecht's study on HAMR objects revealed that the inclination and eccentricities are both highly effected by SRP, and to a lesser extent solar and lunar gravitational forces, with eccentricities ranging from 0.05 to 0.6 [46]. Schildknecht's study also showed that HAMR objects also tend to have a mean motion of about one revolution per day about their center of gravity [46].

To acquire more information on sizes, shapes, and possibly materials of these HAMR objects, light curve data was gathered. The light curves revealed a high variability of information indicating both periodic and random variations of several orders of magnitude over a few minutes of observations to constant brightness over durations greater than 10 minutes. The data also demonstrated that the behavior of any single object could completely change from one observation to the next. These results indicate randomly tumbling objects with complicated shapes [46].

Finally most HAMR objects seem to have a relatively constant area-to-mass ratio over time; however, some variations in this ratio were recorded and could change by up to a factor of four over a few days [46].

Schildknecht concluded that additional observing techniques would be beneficial to determine the material and the nature of HAMR objects, suggesting photometry, spectroscopy, and radar as potential methods. He also indicated that continuous tracking of

a subset of high-area-to-mass ratio objects would be mandatory to ensure the space object catalog was current [46].

#### ***2.3.4 Resolving Uncorrelated Tracks***

Coffey et al. make it clear that correlating tracks of known space objects that are not associated with an entry in the space catalog is a complex and difficult problem [47]. Currently the DoD catalogs more than 22,000 orbiting objects, including active and inactive satellites, rocket bodies, and debris [48]. These numbers are expected to grow exponentially when the SSN is upgraded in the future allowing the detection and tracking of even smaller objects. These smaller objects are already on orbit, but undetected by the current sensors. When future sensors locate them they will start as UCTs that need to be resolved compounding the current problem [7]. A recent report from the NRC found that:

“Advanced data association methods such as multiple hypothesis tracking will produce a significant improvement in the automation of the space surveillance system, including the sensor processing. The algorithms should be adapted to high-performance computing and advanced computer architectures and should make use of kinematic, feature, and nontraditional data” [6].

Many innovative methods attempting to resolve UCTs depend on sensor data association techniques. Section 2.6.3 discusses UCT resolution techniques and Section 2.3.5 discusses data association methods.

#### ***2.3.5 Data Association and Information Fusion***

Data association and fusion of information are fundamental problems in tracking multiple space objects. Data association becomes a problem when a collection of reports, data measurements and sensor tracks, need to be analyzed to determine which reports originated from the same object and which belong to uncorrelated tracks or false reports. Once this correlation has been accomplished, information about a single object from one or more sources needs to be combined to improve our understanding of the object [6].

Current algorithms are no longer sufficient to handle the number of UCTs, and future demands will far exceed current demand. Furthermore, future demands will require new data association and nonlinear estimation algorithms, advanced computer architectures, orbit propagation, and orbit determination.

Currently AFSPC uses a non-statistical fixed gated data association method known as the Report Observation Association (ROTAS). The ROTAS system works well with widely spaced objects against a benign background. Recent developments have made space more congested with LEO breakups and clusters of objects at GEO that require new data association methods because the objects can be closely spaced and the association is often ambiguous [6].

According to the NRC, a good space object association process should make use of other pieces of data as well: Radar cross section, intensity, and other features could be used to identify an object which will help correlate it to an orbit. Data not traditionally used for tracking space objects (multi-band photometry, radiometry, etc.), should also be considered [6].

A UCT poses an even greater challenge. The problem of associating a UCT from one sensor to another sensor, or even the same sensor after several orbits, is quite difficult. This challenge defines the data association problem today, considered a central problem in multiple-target tracking. The problem is further complicated by miss-association or cross-tagging of objects [6, 49–51].

The data association problem for space surveillance stems from the need to partition sensor tracks into valid objects, and false reports or UCTs. Algorithms for solving data association problems are generally divided into two classes: single-frame and multiple-frame methods [6].

#### ***2.3.5.1 Data Association: Single-Frame Methods***

Single frame methods include Nearest Neighbor (NN) and Global Nearest Neighbor (GNN). Both NN and GNN work well in light clutter and for reasonably spaced objects<sup>5</sup>. The method currently used by AFSPC is GNN. GNN is a two dimensional assignment problem which attempts to relate all objects in a sensor frame to known objects expected to be present. Unfortunately in the current crowded space environment single-frame methods fail to appropriately identify and tag RSOs leading to an increase in UCTs [6].

#### ***2.3.5.2 Data Association: Multi-frame Methods***

Multiple-Hypothesis Testing (MHT) mitigates miss-association or cross-tagging by holding difficult association decisions in abeyance until additional information is available, effectively improving current decisions by changing past decisions [6].

Although MHT methods have been around for the past 30 years, they have only started to be used in observation systems in the last decade. The development of similar advanced association methods for SSA will potentially address several challenges. MHT methods can be adapted to advanced computer architectures, with a focus on control of the computational costs for these architectures. All available data including kinematic, feature, and nontraditional data (see Section 2.3.5) is required to disambiguate the association process when needed. Association methods themselves should be adapted to the complexity of the problem. For widely spaced objects simple methods such as NN can be used, whereas for LEO breakups and GEO clusters, a full MHT algorithm may be required [6].

Blackman's survey of MHT provides an excellent overview of the basics of MHT. As visual representation of the concept of MHT tends to be the best way to explain it, Figure 2.1 is used to show the predicted locations of two targets and three observations made when the targets were predicted to be present. With MHT any two pairs could be the

---

<sup>5</sup>Reasonably spaced objects at GEO are typically separated by at least 1 km

correct targets, for example  $O1$  and  $O2$  represent a single hypothesis, called  $H1$ . Once the observations are made there are 10 possible hypotheses. These hypotheses range from the most likely scenario,  $P2$  belonging to  $O2$  and  $O1$  and  $O3$  being attributed to  $P1$ , to the least likely, all three observations are false alarms [52].

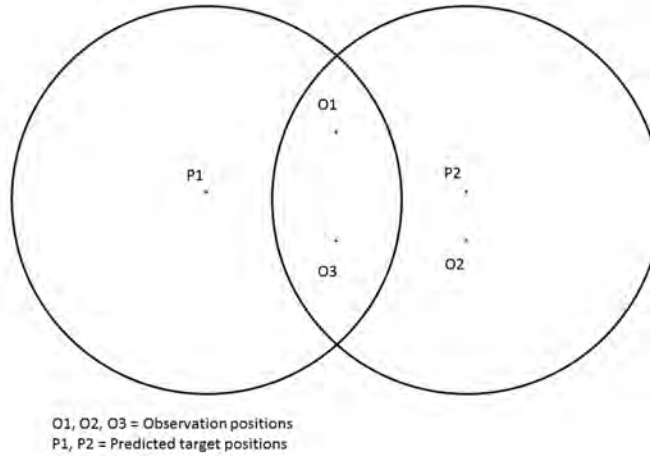


Figure 2.1: Typical data association conflict: prediction variance vs. observed objects

After Singer, Sea, and Housewright introduced the concept, Reid developed his original algorithm which is implemented as a sequential k-best, two dimensional assignment algorithm. Reid defined a systematic approach for evaluating multiple data association hypotheses in a clutter or false alarm background [49, 50, 52–55].

Blackman demonstrates how Reid's algorithm works using Figure 2.1. Start by defining tracks  $T1$  and  $T2$  as the predicted positions  $P1$  and  $P2$ . Then, define all newly formed tracks, such as  $T3(P1, O1)$  from the association of  $P1$  with  $O1$ , for other observation and track combinations to ensure all hypotheses are accounted for. Finally, Reid also defines  $NT1$ ,  $NT2$  and  $NT3$  as the new tracks initiated from  $O1$ ,  $O2$ , and  $O3$ . Now we can define all 10 hypotheses:

$$\begin{aligned}
H_1 : & T1, T2, NT1, NT2, NT3 \\
H_2 : & T3, T4, NT3 \\
H_3 : & T3, T5, NT2 \\
& \vdots
\end{aligned}
\tag{2.1}$$

Compatible tracks are defined as those with no common observations and MHT hypotheses are made up of sets of compatible tracks. Reid's algorithm approach also carries over hypotheses from previous scans. When a new scan is performed, the data are combined and consolidated into only the hypotheses that meet the compatibility constraint [52].

According to Blackman a track-oriented MHT approach recomputes all hypotheses using newly updated tracks after every scan. Track-oriented MHT then discards the hypotheses formed on the prior scan after performing the update to create new hypotheses. This method prevents the loss of data, as the tracks that are maintained contain all the relevant statistical data, but does not maintain prior data beyond the current scan. Blackman prefers this approach when there are many track observations (clustered environment) because unlike other methods, it prevents hypotheses from growing exponentially. Alternative methods surveyed by Blackman can be found in Reference 52.

## 2.4 Astrometry Techniques

The science of astrometry aims to accurately define positions of objects in reference to the celestial sphere [33]. To accomplish these measurements in CCD image plate coordinates, pixel locations corrected for distortion must be determined. If sky coordinates are available for known stars in the image they can be used to determine the plate coordinates. These plate constants can then be used to estimate the location of objects with unknown sky coordinates [33]. Alternatively if camera parameters are known then a camera model may be used to determine the real-world position.

Astrometry is practiced by astronomers to define the location of newly discovered objects so that they can be accurately located by others. CCD images can be utilized to obtain precise sky coordinates of celestial objects. Other benefits of this method include the ability to accurately determine the focal length of the telescope used for taking the image, as well as the images precise scale and orientation [33].

The use of astrometry to find moving objects is also well understood. Techniques have been developed for finding asteroids and comets and then computing their orbit around the sun [33]. For man made satellites, the same dim object detection techniques can be used to locate Earth orbiting satellites.

Chapter 3 focuses on the comparison of three methods of astrometry for angle data extraction: affine transform astrometry, the pinhole camera model, and the intrinsic parameters camera model, also known as the intrinsic parameters calibration model. Each method has its own requirements outside of the image, and will be discussed in detail in Sections 2.4.1 through 2.4.3. Both the pinhole camera model and the affine transform method have been successfully used in astrometry in the past to track RSOs [13, 33]. The validation and verification of each model is covered in Appendix B.

#### ***2.4.1 Affine Transform Astrometry***

The first method considered for extracting satellite position information from an image will be the application of an affine transform. Requiring only the proper identification of stars in the image, and number of pixels in the frame, affine transform astrometry requires the smallest amount of data of any of the considered methods [33].

To use the affine transform method, start by measuring  $(x', y')$  locations for each reference star and collect the corresponding look angles; right ascension and declination. Then compute plate constants, transformation parameters between image pixel coordinates and sky coordinates ( $\alpha$  and  $\delta$ ), for the image. Once plate measurements are known, measure



the  $(x', y')$  position of the target objects. Finally, compute the coordinates of the target object. Each of these steps is outlined below.

#### **2.4.1.1 Standard Coordinates**

The standard coordinates used in astrometry assume that an image of the sky contains a small enough FOV that the spherical surface can be treated as a plane. Thus, the standard coordinate plane is defined as a plane tangent to the celestial sphere at a point  $(\alpha_0, \delta_0)$  on the sky. According to Berry, “the X-axis is aligned with right ascension  $(\alpha)$ , the Y-axis is aligned with declination  $(\delta)$ , and the origin lies at the point of tangency.” Therefore it is straightforward to compute the location  $(X, Y)$  of any star with a known right ascension and declination as demonstrated in Equations 2.2 and 2.3 [33]:

$$X = \frac{\cos(\delta) \sin(\alpha - \alpha_0)}{\cos(\delta_0) \cos(\delta) \cos(\alpha - \alpha_0) + \sin(\delta_0) \sin(\delta)} \quad (2.2)$$

$$Y = -\frac{\sin(\delta_0) \cos(\delta) \cos(\alpha - \alpha_0) - \cos(\delta_0) \sin(\delta)}{\cos(\delta_0) \cos(\delta) \cos(\alpha - \alpha_0) + \sin(\delta_0) \sin(\delta)} \quad (2.3)$$

These standard coordinates are useful as they mimic the formation of images on the CCD imager [33]. Thus the left hand side of Equations 2.2 and 2.3 are the same as the left hand side of Equations 2.4 and 2.5 allowing for the determination of the plate constants.

#### **2.4.1.2 Finding Plate Constants**

In an ideal image the  $X$  and  $Y$  axes used in measuring the image would be perfectly aligned with its right ascension and declination axes. The exact center of the image  $(\alpha_0, \delta_0)$  would be given. Under ordinary circumstances, however,  $(\alpha_0, \delta_0)$  will not be perfectly at the center and the  $X$  and  $Y$  axes of the detector will be rotated through some angle, as seen in Figure 2.2. These offsets will be automatically compensated for when the mathematical transform from standard to plate coordinates occurs [33].

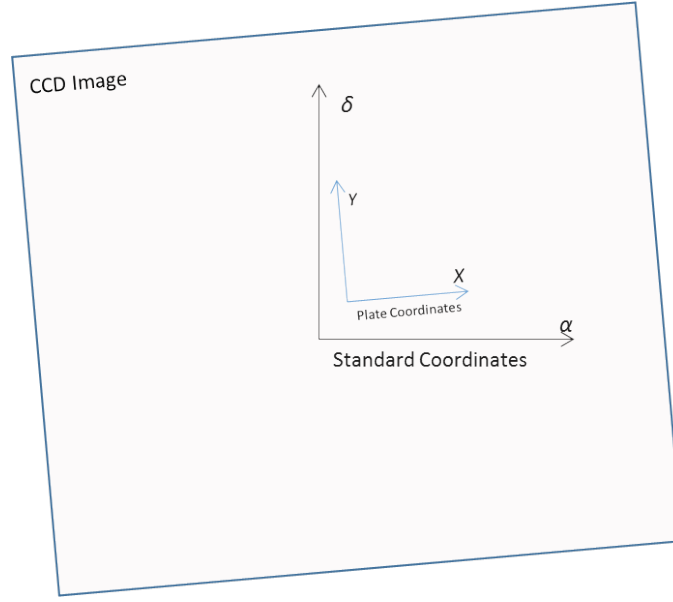


Figure 2.2: Plate coordinates vs. standard coordinates

Offsets between the axes can be caused by displacement, rotation, tilt, and inaccurate pixel dimensions at the CCD's operating temperature. Instead of trying to account for every possible optic and sensor combination in use, Berry states that it is possible to write the relationship between two coordinates systems as a general affine transformation as shown in Equations 2.4 and 2.5 [33]:

$$X = ax' + by' + c \quad (2.4)$$

$$Y = dx' + ey' + f \quad (2.5)$$

With  $X, Y, x', y'$  as previously defined and terms,  $a-f$  the 'plate constants'. The plate constants can be determined empirically from the CCD image itself by measuring the  $x', y'$  location of a minimum of three reference stars, and solving the resulting linear equations for the six unknowns using six independent equations as shown in Equation 2.6 [33]:

$$\begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ Y_1 \\ Y_2 \\ Y_3 \end{bmatrix} = \begin{bmatrix} x'_1 & y'_1 & 1 & 0 & 0 & 0 \\ x'_2 & y'_2 & 1 & 0 & 0 & 0 \\ x'_3 & y'_3 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x'_1 & y'_1 & 1 \\ 0 & 0 & 0 & x'_2 & y'_2 & 1 \\ 0 & 0 & 0 & x'_3 & y'_3 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \end{bmatrix} \quad (2.6)$$

If more than three reference stars are available in the image they can be used to improve the initial calculation of the plate constants by performing a least-squares solution to produce the most probable solution to an overdetermined set of equations [33].

#### 2.4.1.3 *Extracting Position Coordinates from Streaks*

Once the plate constants are known, the position of the object in the image can be determined from its  $(x', y')$  plate coordinates utilizing Equations 2.4 and 2.5 [33]. Finally to convert the standard coordinates of the object back to right ascension and declination use Equations 2.7 and 2.8<sup>6</sup> [33]:

$$\delta = \arcsin \left( \frac{\sin(\delta_0) + Y \cos(\delta_0)}{\sqrt{1 + X^2 + Y^2}} \right) \quad (2.7)$$

$$\alpha = \alpha_0 + \arctan \left( \frac{X}{\cos(\delta_0) + Y \sin(\delta_0)} \right) \quad (2.8)$$

#### 2.4.2 *Frontal Pinhole Camera Model*

An alternative astrometry method is to apply a pinhole camera model to the provided image [13]. This model assumes that all light rays are forced to go through the optical center which minimizes optical distortions. A well-focused system designed such that distortions in the image are negligible effectively becomes a pinhole camera [56].

For the purpose of the comparison performed here the image plane will be placed at a location equal to the distance of the focal length in front of the origin (optical center),

---

<sup>6</sup>The two quadrant inverse tangent is used in Equation (2.8) and the result is added to  $\alpha_0$  to provide  $\alpha$ .

along the  $Z$  axis, as shown in Figure 2.3. This placement defines the conventional frontal pinhole camera model [56].

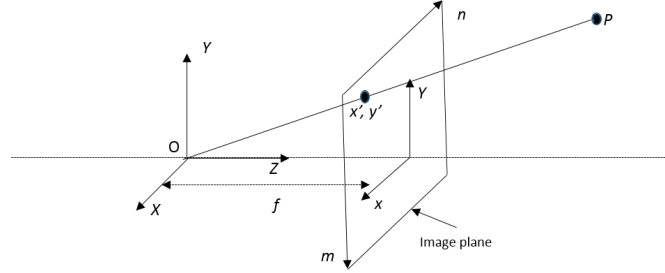


Figure 2.3: Frontal pinhole imaging model

When the pinhole camera model is used, the ideal perspective projection can relate the image of an object to its physical coordinates as shown in Figure 2.3 [56]. The pinhole camera model, provided in Equation 2.9, can be used to convert the pixel location  $(x', y')$  for the observed RSO to an  $[X \ Y \ Z]^T$  position vector.

$$\lambda \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} f' & 0 & O_x \\ 0 & f' & O_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (2.9)$$

Where  $\lambda$  in Equation 2.9 is a normalizing scale factor. As Figure 2.3 shows the sensor's  $[X \ Y \ Z]^T$  plane's origin is along the optical axis, marked by the  $x$  and  $y$  axis origin, placing it in the middle of the image. The  $[X \ Y \ Z]^T$  sensor frame  $X$  and  $Y$  components are just shifted versions of the pixel rows and columns in pixel units, as provided by Equation 2.10, where  $N$  and  $M$  are the number of horizontal and vertical pixels of the CCD.

$$\begin{bmatrix} O_x \\ O_y \end{bmatrix} = \begin{bmatrix} \frac{N+1}{2} \\ \frac{M+1}{2} \end{bmatrix} \quad (2.10)$$

The downrange vector component ( $Z$ ) is determined by the sensor's focal length, in pixels, as derived from the relationship shown in Figure 2.4 [13]. In Figure 2.4,  $\theta$  is the angle at which light from the edge of the FOV passes through the origin of the coordinate system. The prime symbol is used to indicate virtual measurements inside an optical instrument [13, 57]. From Figure 2.4 you can see that  $\theta'$  equals half the angle that defines the FOV. Therefore the focal length can be calculated in terms of pixels using Equation 2.11<sup>7</sup> [13, 56].

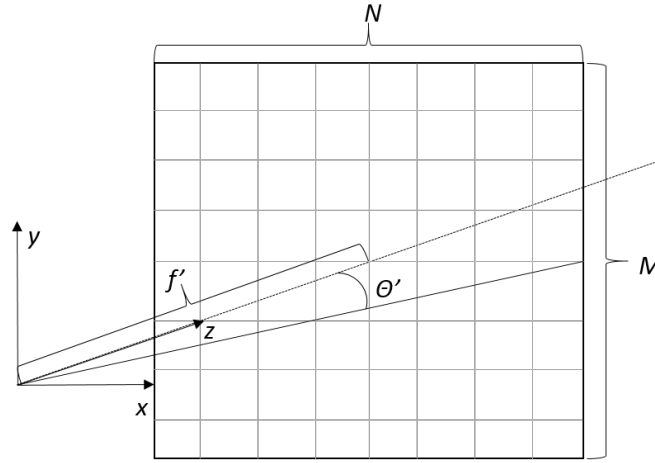


Figure 2.4: Virtual focal length ( $f'$ ) as derived from field of view ( $\theta$ )

$$f' = \frac{N}{2 \arctan\left(\frac{\text{FOV}}{2}\right)} \quad (2.11)$$

All of the variables required for Equation 2.9 have now been fully defined from pixel coordinates provided by an  $N \times M$  image. A point located at pixel coordinates  $(x', y')$  in the image frame can be transformed into a position vector using Equation 2.12 [13].

<sup>7</sup>The two quadrant inverse tangent is again used in Equation (2.8) to define the length of  $f'$

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} x' - O_x \\ O_y - y' \\ f' \end{bmatrix} \quad (2.12)$$

### 2.4.3 The Camera with Intrinsic Parameters

If in addition to the angle to the principal point and the FOV, the aspect ratio and skew factor are also known, the intrinsic parameters or calibration matrix can be used to solve the astrometry of an image. Like the frontal pinhole camera model, this method defines the pixel  $(x', y')$  where the desired object is located and then transforms it to a position vector, as shown in Figure 2.5.

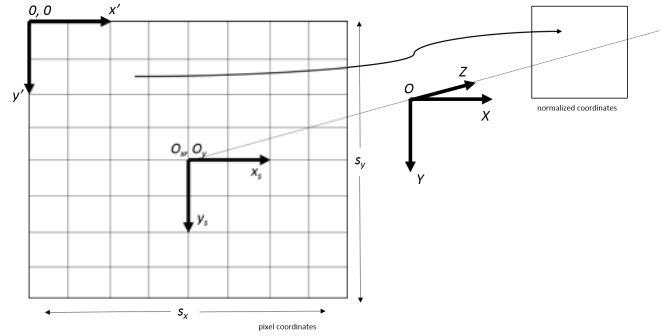


Figure 2.5: Transformation from pixel coordinates to  $[XYZ]^T$

The intrinsic parameters camera model given in Equation 2.13 can be used to determine the real world position of an object in a CCD image. As long as the intrinsic parameters; aspect ratio ( $\eta$ ), skew factor ( $s_e$ ), and focal length ( $f'$ ) are known, this model can define the pixel location of an object in a normalized reference frame creating a position vector [56].

$$\lambda \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} f' & s_\epsilon f' & O_x \\ 0 & \eta f' & O_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (2.13)$$

Where  $\lambda$  in Equation 2.13 is once again a normalizing scale factor and the skew factor is proportional to  $\cot(\epsilon)$ , where  $\epsilon$  is the angle between the  $x_s$  and  $y_s$  image axes. In most systems  $\epsilon$  is very nearly  $90^\circ$  so  $s_\epsilon$  will be assumed equal to zero [56]. The inverse process to determine the real-world location of an RSO given the focal plane pixel location is given in Equation 2.14.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} x' - O_x \\ \frac{y' - O_y}{\eta} \\ f' \end{bmatrix} \quad (2.14)$$

In addition to the intrinsic parameters, to use this approach knowledge of the extrinsic parameters;  $\alpha_0$  and  $\delta_0$  defining the boresight look angle of the system is required to correctly transform the image to the ECI coordinate system [56]. Once a position vector in the ECI reference frame is available, a spherical coordinate transformation can be performed to yield the look angles from that sensor to the object, successfully extracting the angles data from the image allowing for orbital determination or analysis.

#### **2.4.4 *astrometry.net***

Lang's recently developed *astrometry.net* software performs image calibration for astronomical images through geometric hashing. This tool identifies the background star field and returns the principal point, look angle, and the required information to calculate the FOV of the image, allowing for the determination of the virtual focal length [13, 56, 58]. This information allows for the intrinsic parameters ( $s_\epsilon$ ,  $\eta$  and  $f'$ ) of Equation (2.13) to be determined, yielding the required conversion from a real-world position vector to pixel space. The extrinsic parameters ( $\alpha_0$  and  $\delta_0$ ) are also returned [58].

The *astrometry.net* software has previously been used for SSA research in the tool *GEODETICA* at AFRL and by Moomey at AFIT to resolve the astrometry of an RSO in an Electro-Optical image [59, 60]. The software embeds the astrometry WCS into the returned fits file, using the USNO B catalog<sup>8</sup>, allowing the user to simply read off the right ascension ( $\alpha$ ) and declination ( $\delta$ ) of any pixel, eliminating the need to perform complex calculations to find the  $\alpha$  and  $\delta$  of the satellite streaks. Furthermore, calibration frames (dark, flat, and bias frames) do not impact the result because CCD calibration is more important for light curve data than astrometry [33]. As discussed in Section 3.12 *astrometry.net* will also be used in this work as covered in Appendix A to provide the right ascension and declination of the identified RSO streak.

## 2.5 Streak Identification and Isolation Research

Although astronomers use integration techniques to remove streaks from their data, streak isolation in astronomical images for SSA has only recently been seriously investigated. In the last several years, multiple methods have been developed by different organizations to extract streak data from astronomical images [61]. Wallace, for example, has developed an approach using an algorithm for streak detection within a known astronomical collection [62]. Oniga developed a feature-detecting algorithm that both detects and classifies streaks as either RSO, aircraft, or other using decision trees with approximately 98% accuracy [61]. For continuous sequences of unresolved imagery, Sease and Flewelling developed a tool to detect and track RSOs called *GEODETICA*. The General Electro-Optical DETection, Tracking, Identification, and Characterization Application (GEODETICA) tool uses phase congruency, edge detection, and a Kalman filter-based multi-hypothesis point tracking framework to process optical data and provide both orbital and light curve data in an automated fashion [59]. *GEODETICA* uses

---

<sup>8</sup>The USNO B catalog is an ECI J2000 referenced catalog, properly placing all objects in the image as if they had been observed from the center of the earth at 1200 on 1 January 2000.



*astrometry.net* to resolve the astrometry of the RSO in the image, returning the WCS right ascension and declination.

Sease has demonstrated that the matched filter process can be applied to telescopes tracking near Earth objects that appear to curve in the image plane<sup>9</sup> by first removing curvature from star streaks, leading to homogenous streak geometries despite the location of the star in the image plane. The Sease process depends on re-sampling the image about the axis of rotation, and returning the image in either polar or spherical coordinates to provide an image where all the star streaks are uniform overcoming the decrease in Signal-to-noise Ratio (SNR) caused by the angular rate of the sensor [63, 64].

The most directly applicable research to the presented work herein has been published by Levesque, who uses a series of background estimations and removal techniques, followed by an iterative matched filter to detect RSO streaks completely contained in the image frame. This method can detect  $V_{mag} = 15$  using a 0.35-m telescope. Unfortunately, the matched filter requires the use of a TLE, limiting this method to catalog maintenance and tracking of known satellites [65].

Furthermore, Nikolaev et al. showed that complete application of Levesque’s model is not always required. Applying only a subset of Levesque’s method allows for the extraction of brighter objects from Electro-Optical data [66]. Similarly, Appendix A covers how parts of Levesque’s method is applied for streak extraction in this work.

## **2.6 On-going Parallel Research**

### ***2.6.1 Air Force Research Laboratory Research Efforts***

Currently AFRL is developing new MHT orbital determination tools including SADIE and Ananke in parallel to AFSPC’s acquisition of the new Joint Space Operations Center Mission System (JMS) [67, 68]. These efforts are striving to make the best possible orbital

---

<sup>9</sup>The curved appearance is caused by the Earth’s rotation, as the Earth rotates the celestial sphere remains stationary causing curved star streaks in images with sufficient exposure, or when tracking motion through image stacks using *GEODETICA*.

determination tool set available to future space operators and intend to address many of the concerns presented in [6]. This work is intended to complement those ongoing efforts with a focus on observations made by non-AFSPC (external) sensors. An orbit determination model, developed by Goff and covered in Appendix C, will be used to show the applicability of the methods discussed. This model will be limited to the processing power routinely available at AFIT and will not be as robust as the expected results of future systems. The results of the research effort presented in Chapter 4 are expected to improve when SADIE or JMS are used for orbital determination.

### ***2.6.2 Search and Determine Integrated Environment***

The Search and Determine Integrated Environment (SADIE) suite consists of three development efforts, Auto-Search and Determine (SAD), covariance based track association, and SAD like optical processing, designed to resolve UCTs. Auto-SAD was developed by the Naval Research Laboratory (NRL) to link together SAD and several legacy tools requiring manual implementation to resolve UCT [7]. SAD was originally developed for the Air Force (AF) Space Surveillance System (SSS) fence processing and requires position observations and executes in a parallel computing environment to determine orbital elements from a set of unknown objects using a static algorithm and fixed gates for correlation [69]. Auto-SAD is an automated system designed to replace this manually-intensive process [7].

### ***2.6.3 Resolving UCTs with Search and Determine Integrated Environment***

Tracks remaining after the Special Perturbations (SP) Satellite Identification (SID) correlation have been performed are input as UCTs into SADIE. Once SADIE has received the UCT it compares it against select candidates from the space catalog. If that fails to resolve the UCT, SADIE will generate new candidate ranges from optical processing and an adjustable statistical correlation metric over as many temporally-separated observations as possible. Orbit determination is performed for each hypothesis and if not all observations

are used or fit intervals are inconsistent with sensor performance that hypothesis is pruned. This process is iterative as shown in Figure 2.6 [7].

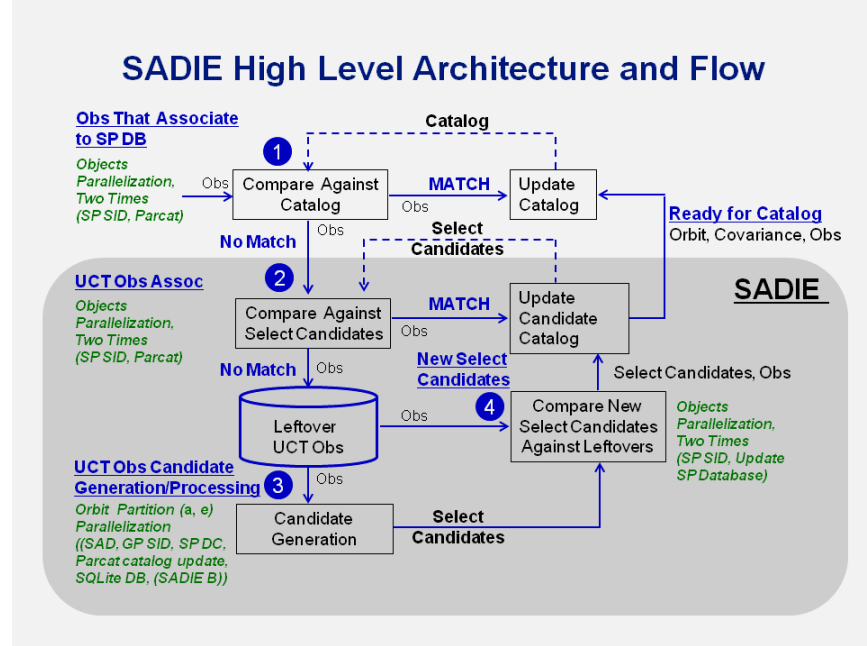


Figure 2.6: SADIE high level architecture [7]

SADIE was recently tested with real-world data at a DoD facility where it demonstrated 98% correlation of objects with six or more observations and 97% correlation with four or more observation tracks. This simulation demonstrated that SADIE could match the outcome of the manually intensive legacy process [67].

## 2.7 Orbit Determination

The focus of the work of Chapters 3 through 4 is on the extraction of angles data from satellite streaks. The presence of both angle data and accurate time stamps, provide the required data for orbital determination using a number of well understood methods including: Laplace, Gauss, and the Double-R method [8, 14].

Orbit determination routines leverage the statistics of RSO orbit propagation and sensor observation to solve the least-squares problem. These routines propagate the orbital solution state, as well as an estimate of the 6-Dimensional Gaussian uncertainty in the state estimate covariance. The Batch Least Squares filter and Extended Kalman Filter (EKF) are commonly used estimation routines [9, 70]. With the increase of computational speed, many favor the UKF as it does not require linearization of the dynamics [71, 72]. The UKF implements the Unscented Transform (UT) that performs nonlinear transformations using “sigma points”. This approach precludes the need for a state transition matrix during covariance propagation, which is often complex for higher-order perturbation modeling and experiences relatively substantial drift after short time frames [8]. Furthermore, recent works have shown that the UKF outperforms the EKF in accurately capturing the estimated error covariance [73]. Finally, since the UKF is a numeric sampling routine, one can implement industry standard high-fidelity orbit propagators and sensor models to estimate states and covariances. When performing orbit determination to validate the usefulness of a result in the following chapters, the method provided in Appendix C will be used.

## 2.8 External Sensor Data

Astronomical observations routinely collect serendipitous satellite streaks in the course of observing deep space [61, 62, 65]. Typically these streaks are simply discarded and without the proper application of *a-priori* data, such as a North American Aerospace Defense Command (NORAD) TLE, many of these streaks would otherwise simply become UCTs.

In the past decade AFRL has published a considerable amount of research on using small commercial telescopes, similar to university astronomy equipment, to accurately track RSOs. These telescopes have demonstrated an accuracy better than one arc-second and the ability to successfully track objects in Highly Elliptical Orbit (HEO) orbits [74, 75]. Recently DARPA’s Orbital Outlook program began purchasing time on amateur

and university astronomy telescopes to actively produce SSA data [76]. Additionally, as covered in Section 2.5 multiple methods have been developed in the last several years by other organizations to extract streak data from astronomical images [61, 65].

Outside of universities, other astronomy sensors of interest include large survey telescopes, such as Pan-STARRS and Large Synoptic Survey Telescope (LSST), along with space-based astronomy equipment operated by National Aeronautics and Space Administration (NASA). In addition to being designed to detect faint objects which also routinely collects serendipitous satellite observations, each of these sensors provides unique advantages [77].

Pan-STARRS for example is co-located with the Maui GEODSS telescope and operated by United States Air Force (USAF) 10% of the time [77]. The USAF operational time means the telescope's operational and calibration status are known. The proximity to Maui means that even when performing its astronomy mission, predicted RSO observations can be removed from the operational mission of GEODSS. If GEODSS is performing fewer routine maintenance observations, then the additional time can be used for operations where higher accuracy is required, such as those missions discussed in Sections 2.3 and 2.2.2.

In contrast to Pan-STARRS, the LSST scope provides geographic separation from all GEODSS locations. This separation minimizes combined observation errors by lengthening the separation in position vectors and providing better geometry to determine radial direction measurements [8]. In addition, the Concept of Operations (CONOPS) for LSST calls for observing the entire western half of the GEO belt once every three days and making its observations publicly available instantly through the world wide web, providing predictable observations of GEO satellites in addition to serendipitous collects of other orbital regimes [78].

Although the wide area survey mission, such as those conducted by LSST and Pan-STARRS are well suited for providing SSA, all astronomical missions can provide

useful data. In addition to astronomy missions AFRL, has several cold staring telescopes that routinely collect partial and full serendipitous satellite streaks. If a method for extracting position data from these streaks was available, this asset would immediately become viable for SSA [25].

## **2.9 Summary**

As this literature review demonstrates, there is a need for improved SSA within the DoD. Several AFRL efforts, including SADIE as discussed in Section 2.6.2, are limited by the number of observations available. Other AFRL initiatives, such as those discussed in Section 2.2.2.3, require considerable time from the handful of high accuracy telescopes available. These needs can only truly be met by increasing the number of data sources available.

In the current fiscal environment the cost associated with building, validating, and maintaining a system precludes the obvious solution of increasing the number of SSN sensors. Other programs such as DARPA's Orbital Outlook program offer alternatives to commissioning new sensors by validating existing sensors and buying time on them for SSA purposes. Astronomical telescopes often collect serendipitous SSA data while performing scientific missions as discussed in Section 2.8 and these observations could provide additional benefits to SSA.

Any of the above sources has the potential to benefit SSA for the DoD. The presence of potential error and distortion sources must be addressed before the data can be integrated into a global SSA concept of operations, as discussed in Chapter 3. Furthermore, as exposure duration for these sidereal tracking systems is often long enough to prevent the capture of complete streaks; alternative methods for using partial streaks need to be developed, such methods are developed in Chapter 4.

### 3. Analysis of Angle Data Extraction Techniques

The science of astrometry aims to accurately define positions of objects in reference to the celestial sphere [33]. Several different approaches exist for obtaining position data from an image: Ma et al. covers a considerable number in Reference 7. This chapter focuses on the comparison of three methods of astrometry for angle data extraction: affine transform astrometry, the pinhole camera model, and the intrinsic parameters camera model, also known as the intrinsic parameters calibration model. Having defined each data extraction method considered, see Sections 2.4.1 through 2.4.3; it is known that each method has its own requirements separate from the image, and these requirements are summarized in Table 3.1 [33, 56]. Both the pinhole camera model and the affine transform method have been successfully used in astrometry in the past to track RSOs [13, 33].

Table 3.1: Requirements for each astrometry method

Method	Known Angles	Focal Length	Other
Affine Transform	$\geq 3$ (stars)	N/A	N/A
Pinhole Camera Model	$\alpha_0, \delta_0$ (degrees)	virtual (px)	FOV
Intrinsic Parameters Model	$\alpha_0, \delta_0$ (degrees)	virtual (px)	$s_x, s_y, s_e$ , FOV

Analysis was performed to determine how each method is impacted by distortions common in optical tracking systems. The analysis seeks to determine the strengths and weaknesses of each method. The detailed analysis of each method should provide answers to the research questions:

1. What is the level of accuracy achieved by each calibration method?
2. Which method best accounts for image distortion?

### 3.1 Generating a Baseline Image

A baseline image, see Figure 3.1, was created using computer simulation. This image consisting of a known star field and four RSO streaks, will represent what we would expect to see from a system that was free of error, imaging satellites with perfectly understood positions. Each star and satellite streak in the image is created by light striking the CCD in a manner that can be modeled as a point spread function (PSF).



Figure 3.1: Simulated baseline image

Mathematically an image taken by an optical sensor takes the form of Equation 3.1 [33]:



$$s(x, y) = k \otimes o(x, y) + n_0 \quad (3.1)$$

where  $k$ , the composite PSF of the system, is the initial impulse response of the system to a point source including any far field distortion effects,  $o(x, y)$  is the undistorted image,  $n_0$  is random additive noise from the CCD itself, and  $s(x, y)$  is the distorted image created by the system. The PSF accounts for diffraction effects of the CCD, the optics in the system, and the atmosphere between the system and the light source [79]. In non-coherent imaging systems, such as astronomical telescopes, the image forming process can be described as a linear system, meaning that superposition applies, i.e. if two objects are imaged simultaneously, the resulting image is equal to the sum of the objects imaged separately. Also, the image of a complex object can be viewed as a convolution of the true object and the PSF [80].

This property of linear systems allows for an ideal image to be distorted in several ways to simulate real-world distortions by convolving of the mathematical model of the distortion with the ideal image. To study the impact noise has on the image, the baseline image will be distorted through computer simulation. Each distorted image will be processed by all three astrometry methods and the displacement of select pixels (stars and satellites) from the baseline image will be measured. RMSE across all displaced pixels will be taken and plotted for each distortion to measure the impact of each type of distortion.

### 3.2 Error Sources that Impact Position

There are several sources of degradation that impact all CCD imaging systems, but not all of these sources impact the recorded position of an object. This research focuses on the sources that impact where the light falls and is recorded in the CCD plane. These error sources include tracking errors, lens effects, jitter, and the atmosphere. Each source, how it was analyzed, and the results are provided below.

### 3.2.1 *Tracking Error*

Although tracking error in images of RSOs can refer to the orbital position error in the TLE for the satellite, in this chapter it is a reference to error in the tracking system. Considerable error in the tracking system can cause a desired object of interest to be missed all together. Small amounts of error however can still result in the capture of the desired object, but will introduce a bias in the process.

To simulate tracking error the baseline image, Figure 3.1, will be processed using the look angle of the star to the left of Ox Vir as the target of the captured observation. As Ox Vir is actually the star being tracked in the image, this incorrect angle simulates an operator incorrectly identifying the right ascension and declination, thereby introducing a tracking error.

### 3.2.2 *Modeling the Effects of a Lens with Aberrations*

The traditional pinhole camera model does not include a lens, and the intrinsic parameters approach assumes a thin lens [56]. Real-world systems often include lenses and these lenses suffer from aberrations that reduce the quality of the captured image by the system as rays from a point source will fail to fall on a common focal point due to the lens through which they pass [33]. If the aberrations are significant enough then ray optics approach is often used to model their effect; however, if the system is near diffraction limited then wave optics is the preferred method [81]. As astronomical optical systems are designed to produce diffraction-limited images, wave optics modeling will be used here [33].

The six most common types of aberrations: defocus ( $W_d$ ), spherical ( $W_{040}$ ), coma ( $W_{131}$ ), astigmatism ( $W_{222}$ ), field curvature ( $W_{220}$ ), and distortion ( $W_{311}$ ) [81]. Each of these aberrations has a different effect on light as it passes through the lens. For example spherical aberration cause rays from sources other than the principal ray to fall ahead of or behind

the focal point, where coma and astigmatism both affect the images away from the optical axis [33].

The lens image simulation approach used here is defined by Voelz in Reference 81 and will only be briefly summarized. First aberrations in the lens causes space variations in the incoherent imaging process meaning that the convolution used in Equation 3.1 cannot be used to model the effects of the lens. Therefore, the superposition integral, provided by Equation 3.2, is required [81].

$$I_i(\hat{u}_0, \hat{v}_0) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} I_g(u, v) |h(\hat{u}_0, \hat{v}_0; u, v)|^2 du dv \quad (3.2)$$

where  $\hat{u}_0, \hat{v}_0$  define the ideal image, and the PSF given by  $|h(\hat{u}_0, \hat{v}_0; u, v)|^2$  is defined by Equation 3.3 [81].

$$h(\hat{u}_0, \hat{v}_0; x, y) = \left| \frac{\sqrt{x^2 + y^2}}{w_{xp}} \right| \exp \left[ -jkW \left( \hat{u}_0, \hat{v}_0; \frac{x}{w_{xp}}, \frac{y}{w_{xp}} \right) \right] \quad (3.3)$$

The value of  $w_{xp}$  represents a multiple of the wavelength of the light collected ( $\Lambda = 0.555 \mu\text{m}$ : the peak of the visible spectrum) and was varied between  $0 \times \Lambda$  and  $10 \times \Lambda$  to account for common lens aberration affect listed above.

To produce the simulated image each point in the image plane was selected, a normalized PSF corresponding to the point was generated and applied to that point. After each point in the image was manipulated, the output image was generated, shown in Figure 3.2. Figure 3.3 also shows the impact of this type of distortion by comparing the centroids of several stars from Figure 3.2 compared to the centroids of the same stars from Figure 3.1. Figure 3.3 show a largely radial distortion pattern that one would expect from a system including a lens and vignetting. The impact of another large error source, the lens, is also apparent in Figure 3.2. For a detailed discussion on this modeling technique please see Reference 81.

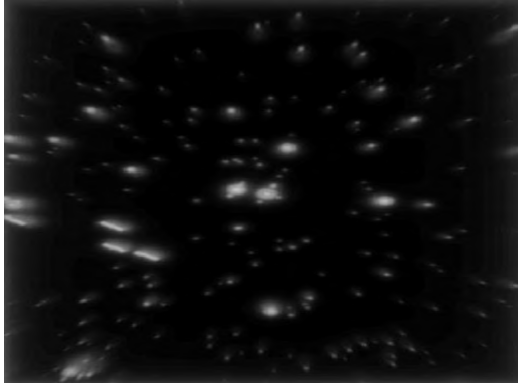


Figure 3.2: Aberrations effects: image

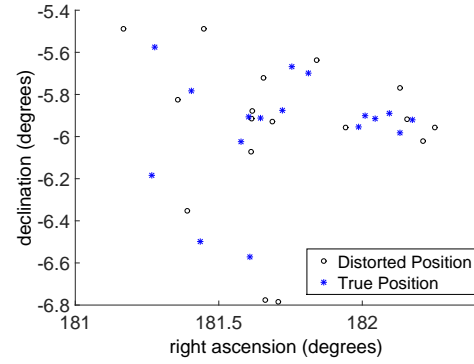


Figure 3.3: Aberrations effects: pixels

### 3.2.3 Jitter

The next source that could impact position of where the light falls on the cells is jitter in the optical system. Jitter, sometimes called “shake,” represents movement in the optical system causing light to fall incorrectly on the imaging plane. Jitter can be introduced by the telescope mount as it tracks objects, by the shaking of the platform the scope is seated on, or even by wind gusts shaking the telescope. In well-designed optical systems the impact of jitter should be minimal [82].

Jitter can be modeled as Gaussian noise and directly effects the optical systems response to the PSF. Assuming the telescope modeled here represents a well-designed optical system, we expect the amount of jitter to represent a small fraction of the FOV [82]. Therefore jitter will be modeled using the modulation transfer function, given in Equation 3.4 and represented by  $k$  in Equation 3.1 [82]. The spatial frequency for the jitter ( $\mu$ ) will be kept below  $\frac{1}{10}$  of the FOV, and the amplitude of the jitter ( $\sigma_{jitter}$ ) will be varied between 0 and  $\frac{1}{2}$ . Convolution of the baseline image with Gaussian filter, using Equation 3.1 and ignoring the CCD noise, effectively simulates the distortion expected from jitter as can be see in Figures 3.4 and 3.5 [82].

$$k(\mu, \sigma_{jitter}) = \exp^{-\frac{1}{2}\mu^2\sigma_{jitter}^2} \quad (3.4)$$



Figure 3.4: System jitter effects: image

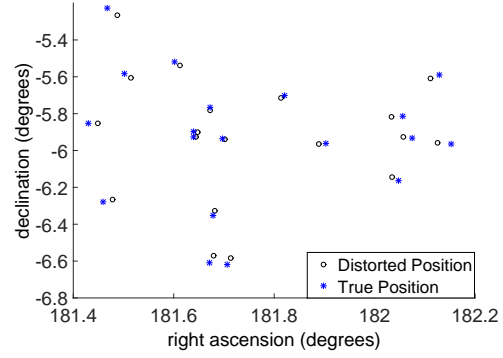


Figure 3.5: System jitter effects: pixels

Figure 3.5 shows that the centroids have been offset slightly from where they appear in the baseline image. This offset is expected from the blurring, visible in Figure 3.4, causes elongation of several stars, thus changing the position of the centroid of the PSF.

### 3.2.4 The Atmosphere

Atmospheric turbulence can have a drastic impact on light passing through it. Common effects include scintillation, best known for causing stars to twinkle, as well as the refraction of light, sometimes leading to multiple copies of a star in an image, known as speckles. In short exposures, less than 1/4 second, these atmospheric differences lead to extremely variable PSF magnitudes in the imaging system. As the PSF of a long exposure experiences turbulence equal to the sum of turbulence experienced in several short exposures across the same time frame, the atmospheric turbulence in longer exposures can be treated as a single random event. Long exposures are often preferred for dim object detection because they have a better SNR [33].

The simulated exposure time used for this phase of the research is 10 seconds. This exposure length was chosen to be longer than 4-5 seconds, considered a minimum exposure by some astronomers, yet short enough to contain complete satellite streaks [61].

Because these long exposures can be simulated as a single random exposure and the atmosphere behaves like Gaussian noise, simulated atmospheric turbulence was created using independent random Gaussian noise with zero mean and normalized variance. This Gaussian distortion was created and convolved with the baseline image using Equation 3.1 and ignoring the CCD noise. This noise effectively blurred the stars and reduced their magnitude in the image.

The results of this noise can be seen in Figure 3.6. Notice that several of the stars are no longer visible as atmospheric interference has increased the noise floor. What is not apparent in Figure 3.6, but does show up in Figure 3.7, is that the atmosphere has also changed where the light is falling on the CCD. Figure 3.7 show that the centroids of the stars have moved from their true positions potentially impacting the perceived position of objects the system is trying to identify.



Figure 3.6: Atmospheric effects: image

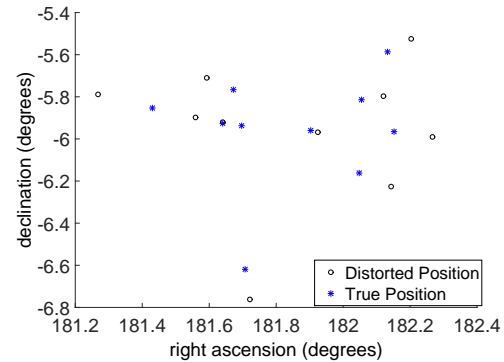


Figure 3.7: Atmospheric effects: pixels

### 3.2.5 Multiple Error Sources

To get an idea of how multiple noise sources effect the ideal image to aid in determining the best overall method, two noise sources where applied to the ideal image simultaneously. The noise sources where atmospheric turbulence and the effects of lens aberrations. These noise sources were chosen because they will be present for any terrestrial based electro-optical system. The results of this stacking of noise sources is displayed in Figures 3.8 and 3.9.

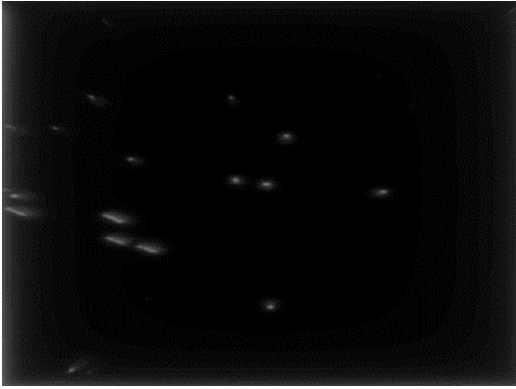


Figure 3.8: Multiple noise sources: image

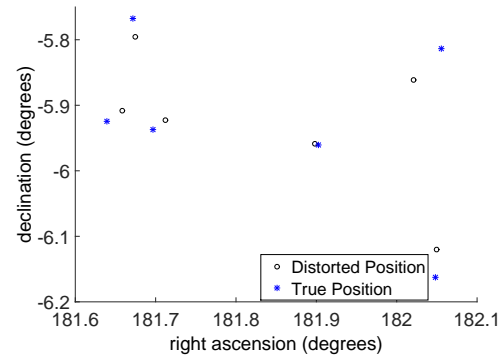


Figure 3.9: Multiple noise sources: pixels

These figures show that the impact of combined noise sources stack just as Equation 3.1 indicates they would. Figure 3.9 shows both the loss of stars from the atmosphere as well as the shift in position caused by both the atmosphere and the lens aberrations.

## 3.3 Real-world Images

As a final test, a series of real-world images taken by the TeleTrak network located at AFIT were used to determine how the astrometry techniques would function on non-calibrated imagery from a real system. The images used contain the major sources of

position error discussed above. They capture Intelsat 1R, as shown in Figure 3.10, as it passed through the FOV of the scope. The scope is in sidereal tracking mode, atmospheric and jitter effects can be observed in the stars, and a lens was present on the system that took the image. The RSO also passed through the system's FOV allowing for the effects of the radial distortion near the edge of the image to be captured in results as well as data from the less distorted region near the principal point.

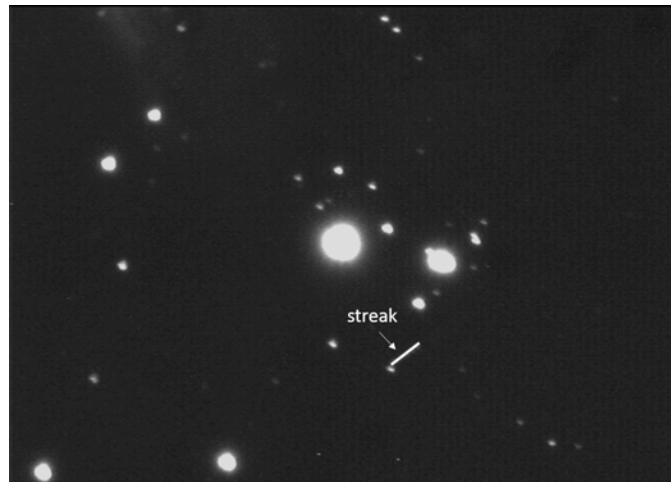


Figure 3.10: A complete RSO streak captured by TeleTrak (contrast enhanced)

### 3.4 Astrometry Methods Performance

The overall performance of each astrometry method can be seen in Figure 3.11 where the RMSE for each method for each noise source is provided. The RMSE reported in Figure 3.11 is derived from the results across all runs of that method. This RMSE analysis allows for an objective comparison of the performance of each method. The RMSE angle and track error is also provided in Table 3.2. It is evident that performing an intrinsic parameters calibration results in the best position vectors when using distorted images. Each individual analysis and the method's performance is further analyzed below.



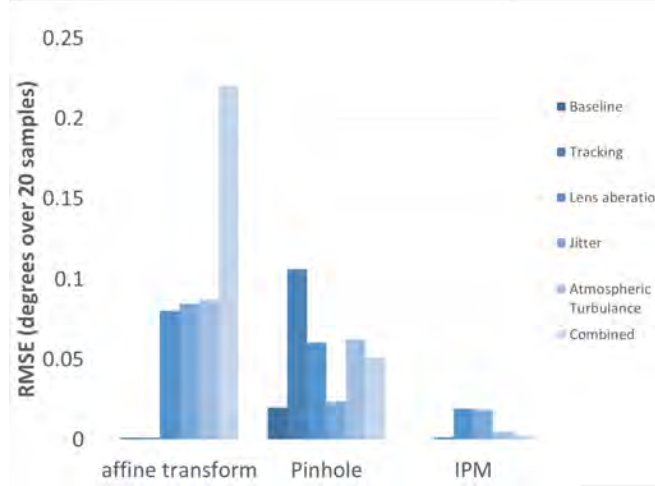


Figure 3.11: RMSE for each method and noise source

Figures 3.12 through 3.17 below provide the results from the five runs of four RSOs for each method to demonstrate that the results are both consistent and representative of the methods considered.

### 3.5 Baseline Image

Because no estimation method is perfect it is expected that each of the methods used here has the potential to introduce error into the position of the observed objects in the frame. The baseline image was processed by all three methods to observe any inherent bias that the methods introduce. The results can be seen in Figure 3.12.

The position that streaks were actually placed on the image, using the process provided in Section 4.5, are also included as “Truth” for comparison purposes. As Figure 3.12 shows both the affine transform and intrinsic parameters calibration astrometry methods introduce little to no bias in the results. The pinhole model does introduce bias as shown in Figure 3.12.

The distortion present in the pinhole camera model is the focal length distortion as expected given the model. Inspection of Equation 2.11 shows that only horizontal focal

Table 3.2: RMSE in degrees and kilometers

Method	Error Source	Angle (degrees)	Track Error (km)
Affine transform	Tracking Error	0.0009	0.6623
	Lens Aberration	0.0804	59.1833
	Jitter	0.0847	62.3632
	Atmospheric	0.0871	64.0734
	Combined Error	0.2204	162.167
Camera/intrinsic parameters	Tracking Error	0.0014	1.0354
	Lens Aberration	0.0194	14.303
	Jitter	0.0183	13.5111
	Atmospheric	0.0048	3.56397
	Combined Error	0.0025	1.87507
Pinhole Camera Model	Tracking Error	0.1063	78.1902
	Lens Aberration	0.0604	44.4374
	Jitter	0.0238	17.5453
	Atmospheric	0.0624	45.9114
	Combined Error	0.0513	37.7804

length is considered in the model. This result is valid given the model makes a square pixel assumption, unfortunately square pixels are not guaranteed in real-world systems and are not present here. As the pinhole camera model does not allow for scaling, some focal length distortion is expected. This focal length distortion will also be present in other results although they may be small by comparison to other error sources.

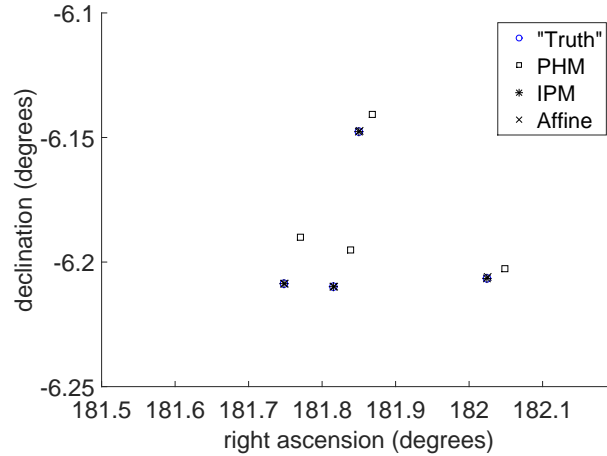


Figure 3.12: RSO position offset introduced by method

### 3.6 Tracking Error Impact

The impact of the induced tracking error can be seen in Figure 3.13. Here a small offset in the true look angle of the telescope vs. the intended look angle leads to error in both right ascension and declination, slightly skewing the results provided by two of the three methods, and drastically altering the results returned by the pinhole camera method.

One method, the affine transform, is virtually unaffected by the tracking error. This result is expected because multiple stars are used to determine the location of the RSOs thus a small mistake in the telescope pointing angle is compensated for. The intrinsic parameters calibration is also able to account for the tracking error, although not as accurately, as shown in Figure 3.13.

The pinhole camera model is the most degraded model returning positions that are off by an RMSE of  $0.11^\circ$ . That  $0.11^\circ$  translates to 36 arc-seconds, or approximately 81 kilometers of along-track error, for a geosynchronous satellite. The intrinsic parameters calibration returned results that were offset by an RMSE of  $0.0014^\circ$ , translating to 5.04 arc-seconds, or approximately one kilometer of track error. Finally the affine transform

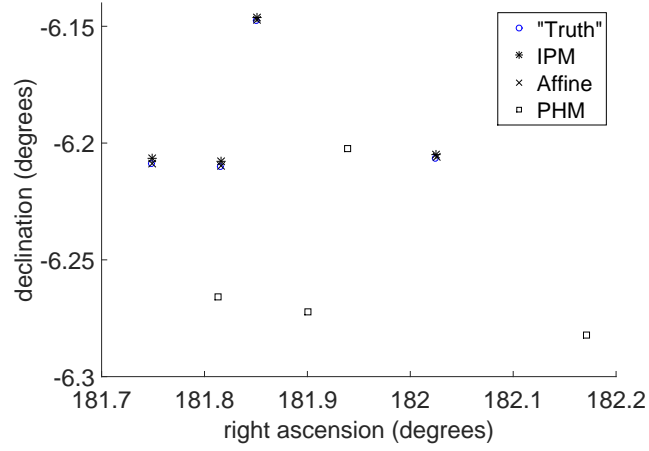


Figure 3.13: RSO position offset caused by tracking error

performs the best with an RMSE of  $0.0009^\circ$ , approximately 3.24 arc-seconds of error, or 66 meters of along-track error.

### 3.7 Lens Aberration Impact Results

The impact of lens aberration is shown in Figure 3.14. Here the intrinsic parameters calibration camera model appears to deal with the lens aberrations the best with its results appearing closest to the true sources of the light, offset by an RMSE of  $0.02^\circ$ . This minimal impact is a result of the fact that the intrinsic parameters account for the displacement of the PSF caused by the lens.

With a lens in place both the affine transform and the pinhole camera model methods move away from truth. This result is not surprising as the affine method struggles with the streak near the edge of the image, and the lens serves to increase edge distortion. Given that the pinhole camera model assumes no lens it is expected that the presence of a non-ideal lens will impact the results provided by the model [56]. The RMSE analysis reveals that

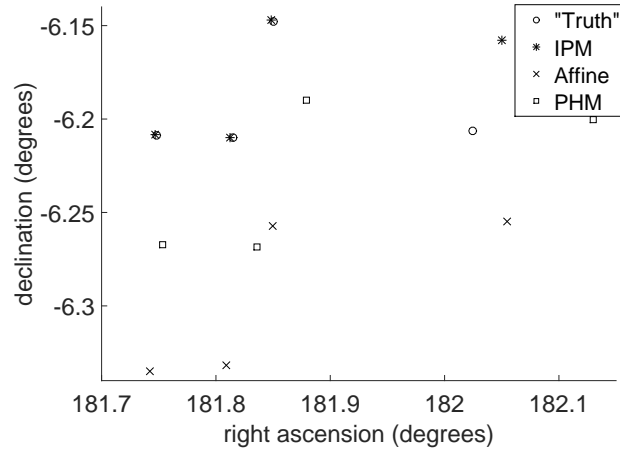


Figure 3.14: RSO position offset introduced by aberrations

the pinhole camera model outperforms the affine transform method with an RMSE of  $0.06^\circ$  vs.  $0.08^\circ$ .

### 3.8 Jitter Impact Results

Jitter impact on the output of an optical system can be seen in Figure 3.15. The RMSE analysis reveals that the intrinsic parameters calibration, with an RMSE of  $0.018^\circ$  slightly outperforms the other methods. This accuracy is also apparent in Figure 3.15 as the intrinsic parameters calibration results are closest to truth. The affine transform method performs the worst in the case of jitter, with an RMSE of  $0.0847^\circ$ , and the pinhole results are close to the intrinsic parameters calibration results with an RMSE of  $0.023^\circ$ .

### 3.9 Atmospheric Turbulence Results

The impact of atmospheric turbulence is displayed in Figure 3.16. Figure 3.16 shows that an intrinsic parameters calibration accounts for the effects of the atmospheric turbulence the best with an RMSE of  $0.0048^\circ$ . Pinhole camera model performs worse than the intrinsic parameters calibration model with an RMSE of  $0.062^\circ$ , and the affine transform

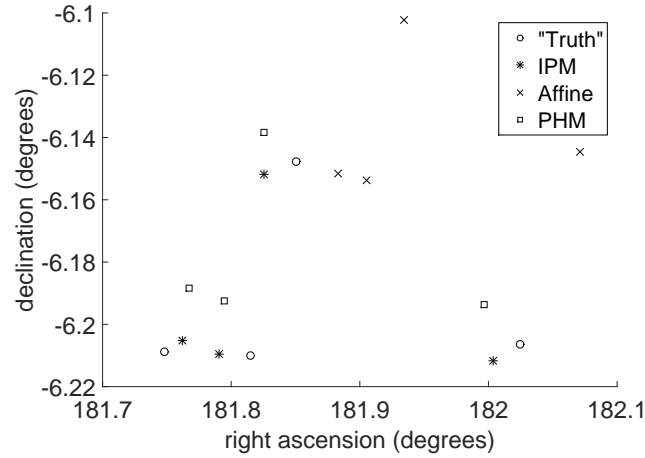


Figure 3.15: System jitter RSO position offset

method handles the atmospheric turbulence the most poorly of the methods considered here with an RMSE of  $0.087^\circ$ . The poor results provided by both the pinhole camera model and the affine transform method is a result of the atmospheric turbulence shifting where the light falls on the CCD as shown in Figure 3.7.

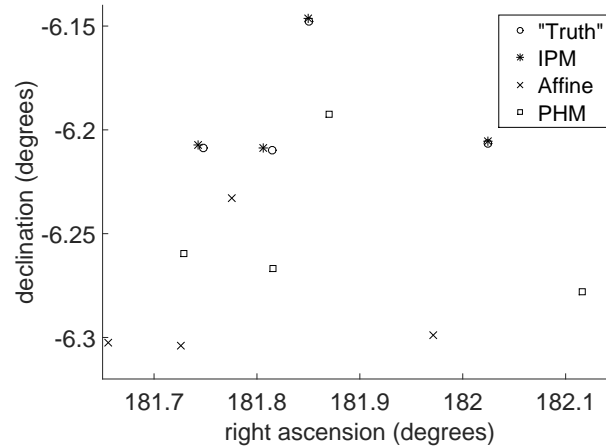


Figure 3.16: Atmospheric turbulence RSO position offset

### 3.10 Multiple Error Sources

As expected, the results of the multiple error sources simulation, shown in Figure 3.17, bears similarities to the results of the two error sources used to create it. The dominant effect is similar to the impact of the lens on the image, but the angles are spread out from the atmospheric turbulence experienced.

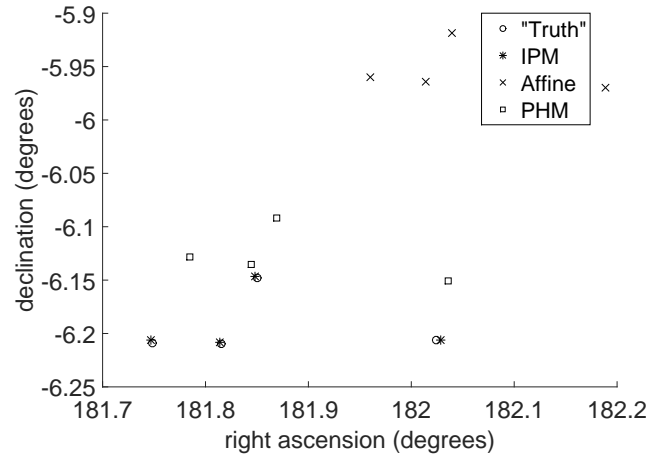


Figure 3.17: RSO position offset introduced by multiple error sources

As shown in Figure 3.17, the camera calibrated with intrinsic parameters model remains closest to truth, with an RMSE of  $0.0025^\circ$ . The pinhole camera model is the next best, with an RMSE of  $0.05^\circ$ , and the affine transform method providing an RMSE of  $0.22^\circ$ .

### 3.11 Real-World System Images

The simulated results provide an understanding of how each type of distortion effects the observed position of an RSO. Indicating that when the intrinsic parameters are available they can provide the most accurate position. Using this knowledge to interpret the real-

world data allows for the determination of how using another method may affect the RSOs recorded position.

Figure 3.18 shows the results of performing the astrometry methods on a series of images taken by a real-world system. The pinhole and intrinsic parameters camera models return very similar results, particularly near the images principal point, indicating that a well calibrated system is responsible for capturing the image [56]. Deviation is noticeable between the pinhole and intrinsic parameters methods near the edge of the image. Overall an RMSE difference from the intrinsic parameters method of  $0.00003^\circ$  right ascension and  $0.0397^\circ$  declination.

The affine transfer method will offset from these other two methods, providing an RMSE difference from the intrinsic parameters method of  $0.64^\circ$  right ascension and  $0.8052^\circ$  declination. The distortion in right ascension is caused by a lack of a good candidate star on the right side of the image and the noticeable change in declination is caused by the weighted centroids of the chosen stars changing pixel location from image to image. Even without these effects, the offset from the intrinsic parameters return is considerable.

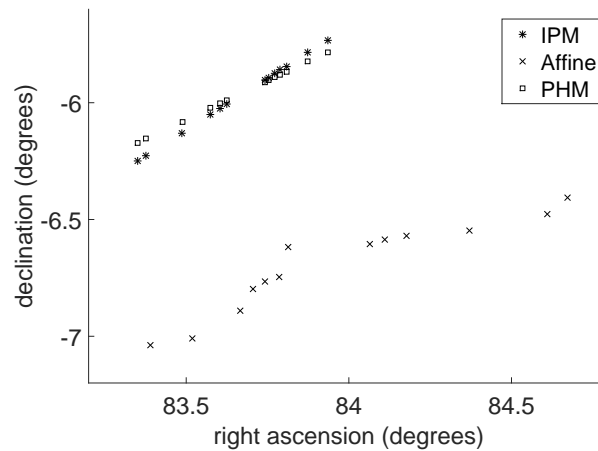


Figure 3.18: Real-world system image detected position



### 3.12 Summary

The analysis given above indicates an uncalibrated image including a streak created by an RSO can be processed to extract angles data. All three methods explored return simulated position data to within  $0.2^\circ$  of the true position without prior calibration. These results are representative of potential serendipitous SSA sources, as most astronomical data sources will provide images that are well defined by a star catalog. Other astrometry methods may also provide equally accurate results and depending on the source of the noise a given method may be preferred over the others.

Sease et al. uses the tool *astrometry.net*, to perform the astrometry process in *GEODETICA* in Reference 59. Lang provides a complete description of the *astrometry.net* geometric calibration in Reference 58 and it outperforms the intrinsic parameters calibration for real-world images, by approximately  $0.5^\circ$ , therefore *astrometry.net* will be used to process the real-world data used in Chapter 4 as outlined in Appendix A.

## **4. Updating Track Data from Partial Satellite Streaks**

This chapter examines the feasibility and utility of performing positional updates for a space object catalog using metric data obtained from streaks gathered by astronomical telescopes. The focus of this work is on processing data from three possible categories: streaks that only enter, streaks that only exit, or streaks that cross completely through the astronomical image. The methodology developed can be applied to any dedicated SSA sensors, or astronomical imagery, to extract data from serendipitous streaks gathered while observing other Resident Space Objects (RSOs) or astronomical objects. The focus of Phase II work will be to answer the following research questions:

1. Can observations of tracks only partially in the image be used to improve SSA?
2. Can observations of tracks passing fully through the image be used to improve SSA?

### **4.1 The Challenge**

Traditional orbital determination uses bounded streaks, along with image time and exposure duration, to determine an accurate observation time and develop position vectors. A bounded streak will be defined in this work to be a streak with complete timing data, provided by the presence of both endpoints clearly in the image, as shown in Figure 3.10. Identification of the center of a streak is used to provide precise timing of the RSO [8, 14].

Streaks lacking one or both end points, as shown in Figures 4.1 and 4.2, do not provide the necessary timing knowledge to determine accurate position and velocity vectors for the RSO using traditional methods. An alternative approach addressing the lack of timing knowledge is needed if partially bounded and unbounded streaks are to be used to perform SSA functions.

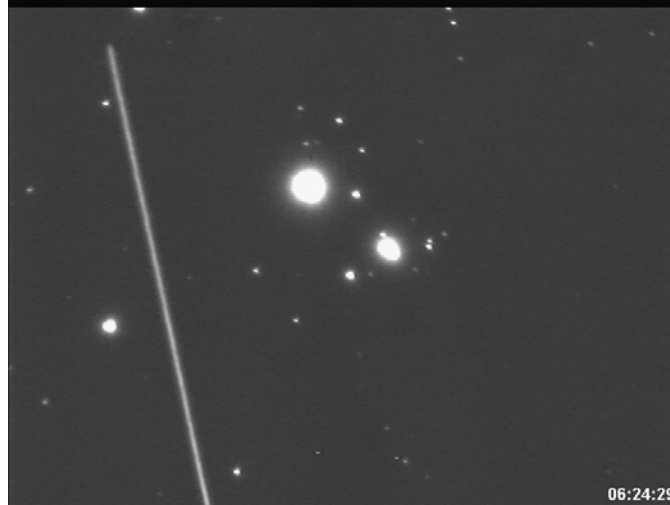


Figure 4.1: A partial RSO streak captured by an optical sensor

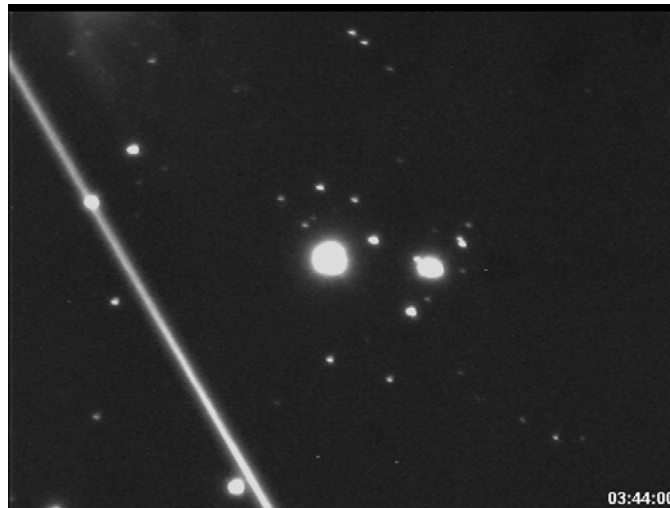


Figure 4.2: An RSO streak passing through a TeleTrak image

To avoid confusion, the term partially bounded streak will be used to describe an image of an RSO with only a single end point visible, as shown in Figure 4.1, while a unbounded streak will refer to a streak that passes completely through the image as shown in Figure 4.2.

## 4.2 Methodology

To determine the usefulness of partially bounded and unbounded streaks two experiments were conducted using observational data collected on Intelsat 1R. The data used was collected by the TeleTrak network as described below. The raw data of two second exposures was used to provide a control data set. These data were also stacked to provide simulated longer exposures and the equivalent streaks that longer exposures provide. In this manner, the same observations could be used to generate both “truth” positions, the centroid of the RSO in a single frame, several images of partially bounded streaks, and one unbounded streak for each night of observations, to serve as experimental groups. Each simulated image and corresponding streak was processed. Comparisons between the results for the control group and the experimental groups are made to analyze performance.

## 4.3 Hardware Setup and Data Simulation

The TeleTrak network has a variety of optical and electromagnetic sensors. Originally designed to track low Earth orbiting satellites, the network has recently started tracking geosynchronous satellites. The data provided for this experiment was a series of two-second exposures, each consisting of 120 stacked frames taken by an astronomical camera attached to an 80-mm telescope on a mount in sidereal tracking mode.

The provided images were then processed to extract the centroid location of each RSO and provide positional data using the traditional method, as discussed in Section 2.2.1. The positional offset of these streaks from the predicted position provides a baseline accuracy to compare against the partially bounded streak data when examining residuals and other orbital data.

### 4.3.1 *Partially Bounded Streak Simulation*

To simulate partially bounded streaks, the two-second exposure images were also stacked to create images similar to Figures 4.3. The stacking of the two-second exposures created the simulation of images with longer exposure times to provide partially bound

streaks. Streaks with only a single end point were created so that the visible edge was present in multiple places in the image as the satellite passed through the FOV. This process allowed for data extraction from both the center and the edge of the image to simulate correctly the real-world captures. Gaps are visible in the streak because the system uses an electronic and not physical shutter, the processing algorithm is designed to ignore a certain amount of space to see a complete streak.

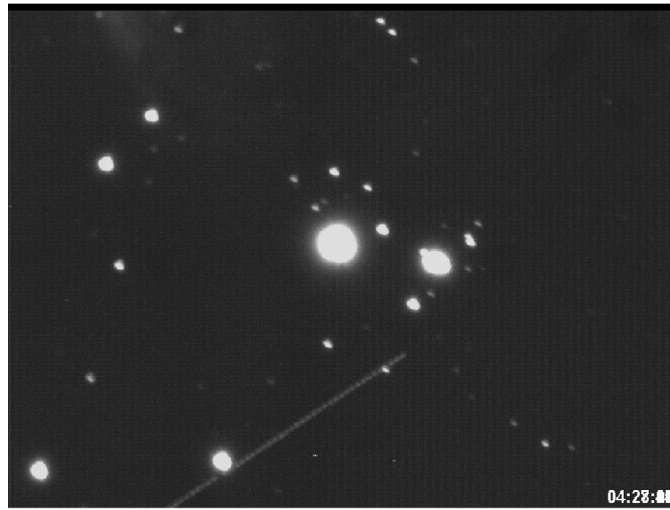


Figure 4.3: A simulated partially bounded RSO streak

#### ***4.3.2 Unbounded Streak Simulation***

Next, the unbounded streaks were created by stacking all of the two-second exposure images from a single night's pass to generate a streak completely through the image, as shown in Figure 4.4. Additional images of the star field were also stacked to lengthen the simulated exposure beyond what was required to create the unbounded streak. The number of additional images was chosen to increase the length of the exposure as much as possible, without capturing a second RSO in the image. Multiple RSOs in the image increases the difficulty of correctly correlating the objects to their TLE and could lead to cross-tagging satellites.

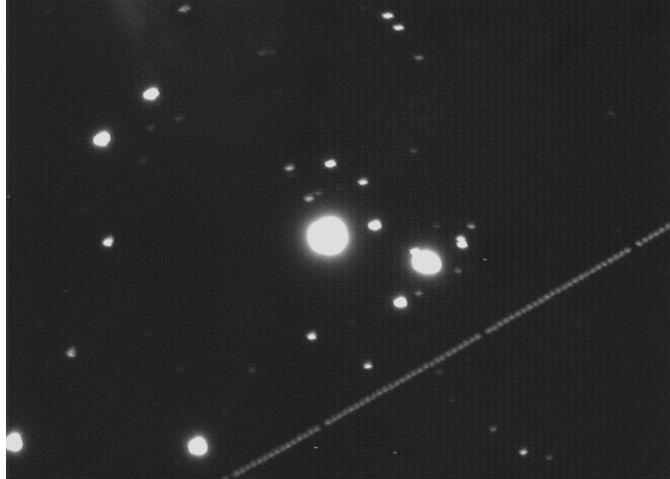


Figure 4.4: A simulated unbounded RSO streak passing through the frame

#### 4.4 Assumptions and Limitations

To determine correctly the position and an associated time of observation of an RSO, several assumptions were made. First, the source RSO is known. Here, three nights of observational data was used to correlate the RSO to a single satellite after the observation was made.<sup>10</sup> Second, the FOV is small enough that the orbital arc can be estimated as a straight-line distance, but large enough so that *astrometry.net* software can successfully resolve the image's astrometry. Finally, the length of the exposure, and its stop time, are known.

#### 4.5 Predicted Position

The results of this work are also limited by the accuracy of the image data provided, the TLE, and the propagator used. To estimate the amount of error expected in the results, a prediction will be accomplished using the TLE retrieved prior to the first night of observations. For these experiments, the TLE will not be updated after the first night, to prevent a catalog update from skewing the results of the experiment. The TLE of the

---

<sup>10</sup>the presence of the streak can also be predicted in advance for some survey missions such as the LSST

observed satellite will be propagated using NORAD's SDP4 propagator, and Systems Tool Kit (STK)'s High Precision Orbit Propagation (HPOP) propagator [83]. Known sources of error will be estimated and modeled to provide a more precise estimated position. This prediction will then be used to create residuals from the recorded observation as discussed above.

Even prior to the first observation it is possible to estimate the amount of position offset caused by the TLE. Accuracy of TLE data for geostationary orbits and specifically Intelsat satellites have been studied in great detail in the past. Chan and Navarro compared the publicly accessible TLE with INTELSAT proprietary orbital elements determined using their own dual station ranging system. The purpose of their comparison was to find out how reliable the TLE was for conjunction detection analysis. After comparing data for a full year Chan found that the orbital differences showed an average of 30-60 km difference in position, when the most recent TLE was propagated to the observational epoch [43]. Further studies have been performed on the accuracies of GEO TLEs in general. Früh and Schildknecht performed a four year study and after evaluating 13 GEO objects found an average of 25 *km* along-track and 10 *km* cross-track difference after propagating the most current TLE to the current epoch. They also found that the error grows the further from the TLE epoch, reaching hundreds of kilometers of track error in just a couple of days [44]. Vallado, published a study in 2007, comparing precise orbital ephemeris data to STK HPOP propagator, and found it more accurately predicted the position of GEO satellites [84].

For these experiments, Intelsat 1R was observed for three consecutive nights by the TeleTrak network of sensors located at the Air Force Institute of Technology. The epoch time on the TLE used was approximately a day before the first expected observation. Based on the work provided in References 43 and 44, it is expected that less than 60 *km* of along-track error and less than 10 *km* of cross-track error will be present due to TLE propagation

using SDP4. For propagation using HPOP an anticipated error of less than 1 *km* of along-track and cross-track error will be present, based on the work provided in Reference 84.

In addition to the position offset caused by the TLE propagation some discrepancy in where the light focuses on the CCD is expected and will have an impact on the position as well. Although the optical system used is diffraction limited no design is perfect, and some distortion is expected. To determine the predicted position of the RSO in the image, the intrinsic parameters camera model will be used. The method is based on the camera model given in Equation (4.1). The camera model is designed to predict the pixel location of an object on a focal plane, given its real-world position [56, 85].

$$\lambda \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} f' & s_{\epsilon}f' & O_x \\ 0 & \eta f' & O_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (4.1)$$

To use this model to predict camera performance requires knowledge of the system's intrinsic parameters ( $f'$ ,  $s_{\epsilon}$ ,  $\eta$ ,  $O_x$ , and  $O_y$ ), provided in Table 4.1, as well as the extrinsic parameters, to define  $\mathbf{R}$  and  $\mathbf{t}$  the required rotation matrix and translation vector to transform the position vector from real-world coordinates to pixel space coordinates, as discussed below. Additionally a Cartesian coordinate position vector for the real-world location will need to be provided.

Table 4.1: Intrinsic and Extrinsic Parameters for the Optical System

Intrinsic Parameters (px)	$f'$	$s_{\epsilon}$	$O_x$	$O_y$	$\eta$	M	N
Value	62888	0	450.5	600.5	1	900	1200

Most star catalogs provide right ascension and declination angles for the positions of stars and most satellites define their position with the most recent TLE. Stellarium does an excellent job of converting the TLE to a right ascension and declination look angle at the



required time, using the SDP4 propagators [86]. When this method is used, conversion to Cartesian direction can be accomplished using Equation 4.2.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} \cos(\delta) \cos(\alpha) \\ \cos(\delta) \sin(\alpha) \\ \sin(\delta) \end{bmatrix} \quad (4.2)$$

Alternatively, STK can be used to provide a predicted Cartesian coordinate vector, using a number of different propagators, as was done for the unbounded streak experiment. Once these Cartesian coordinates are provided, Equation 4.1 can be used to predict pixel location.

The extrinsic transformation required in Equation 4.1 is to assure alignment of the Cartesian coordinate vector and the camera model's reference frame. Using the known principal point of each image, an Euler angle rotation will provide the required transformation, first rotate about the  $K$ -axis by  $\alpha_0 - 90^\circ$  to align the  $X$ -axis with the  $I$  axis and the  $Y$ -axis with the  $J$ . Then, rotate about the  $X$ -axis by  $\delta_0 - 90^\circ$  to bring the  $K$  and  $Z$ -axis into alignment. This rotation is accomplished using Equation (4.3) [8].

$$\mathbf{R} = \begin{bmatrix} \cos(\alpha_0 - 90^\circ) & \sin(\alpha_0 - 90^\circ) & 0 \\ -\sin(\alpha_0 - 90^\circ) & \cos(\alpha_0 - 90^\circ) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\delta_0 - 90^\circ) & \sin(\delta_0 - 90^\circ) \\ 0 & -\sin(\delta_0 - 90^\circ) & \cos(\delta_0 - 90^\circ) \end{bmatrix} \quad (4.3)$$

To verify the model, a star field of 20 stars with known look angles was modeled. The angles were processed to predict the pixel locations of the known stars, given the intrinsic and extrinsic parameters of the camera used. The output was verified by comparing the pixel locations returned to the locations of the stars in a real image taken by the camera. The results of the camera model verification are shown in Figure 4.5, predicted position overlying the image positions, for visual comparison. The overlay of the true position on the returned values is almost an exact match, but residuals are evident in the image. The RMSE of the residuals provides a position offset of 2.73 pixels in the  $x$  direction and 3.4

pixels in the  $y$  direction. Given that the camera averages  $3.28 \text{ arc} - \text{seconds}$  per pixel this offset translates to roughly two kilometers of orbital position error at the GEO belt.

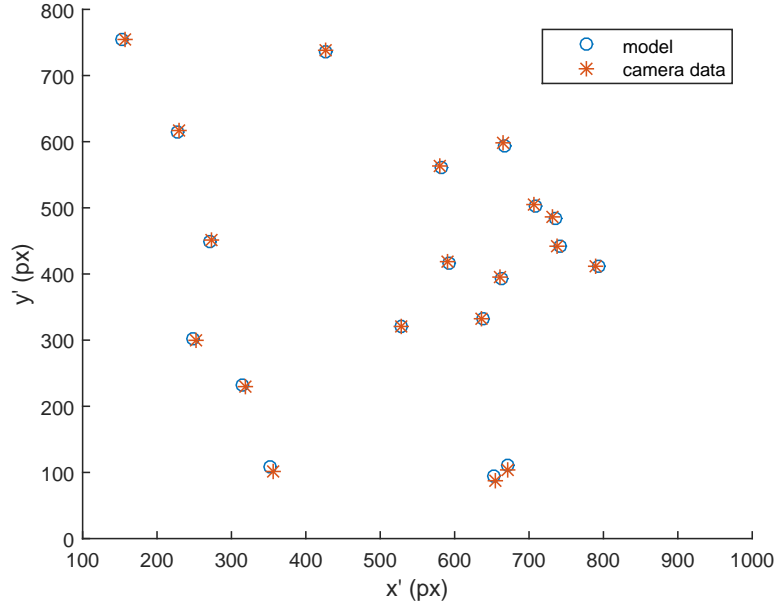


Figure 4.5: Camera model predicted star positions overlying true positions (px)

The last potential sources of position offset are atmospheric distortion and the time data embedded in the header of the image. The impact of the atmosphere is well understood and can be modeled as white Gaussian noise, with the mean atmospheric impact on an optical sensor of approximately  $200 \text{ m}$  [44]. The time stamp for each image is provided by the USNO and is applied to the image, by the network, after integration completes. Discrepancies in the time stamp directly result in position discrepancies. Recent calibration studies on the equipment used have found that the time data can offset the along-track position by as much as  $10 \text{ arc} - \text{seconds}$  [60].

Given the potential error sources discussed above, it is expected that a position offset for the observation of the target RSO will occur. Approximately  $100 \text{ km}$  along-track and

50 km cross-track will result as shown in Table 4.2. The majority of this error is caused by the propagation of the TLE and grows as we move further from the TLE epoch time. Some error is expected from the time meta-data and the remainder, constant error, is caused by the accuracy of our camera model.

Table 4.2: Predicted position RMSE for the Intelsat 1R TLE

	SDP4	HPOP	Time	Sensor	Atmosphere	SDP4 Total (km)	HPOP Total (km)
along-track	60	0.8	2	1.83	0.20	64.03	4.83
cross-track	10	0.8	0	2.28	0.20	12.48	3.28

## 4.6 Extracting Streak Data

To determine the real-world position of the RSO present in the image, a two step process was used. The streak pixel locations are extracted as described below. Then, to determine the real-world location of an RSO given the pixel location provided, *astrometry.net* is used. For the detailed step by step image processing method used see Appendix A.

### 4.6.1 Streak Extraction

Determination of the pixel location of the streak often requires detection of a streak at or near the background noise threshold. To achieve this low SNR detection a modification of Levesque’s method was used. Primarily, first the background was modeled and removed, next the bright stars identified and removed, followed by a Hough transform performed as described in Reference 65. To ensure the Hough transform performed correctly, the processing software was instructed to fill in the dark lines caused by the stacking of images and treat the entire simulated streak as a single line. Applying Levesque’s full method would decrease the required visual magnitude of an RSO for detection, but was not required for the streaks in the data used here.

#### 4.6.2 *astrometry.net*

To resolve the astrometry of the objects in the image, an astronomical calibration of the image was performed. The tool developed by Lang, commonly known as *astrometry.net* was utilized to perform this function [58]. This tool identifies the background star field and returns the principal point, look angle, and overlays the WCS onto the image. The presence of WCS information in the header provides the pixel space to a real-world position transformation to ECI coordinates ( $\alpha^{11}$  and  $\delta^{12}$ ), allowing the pixel location of objects of interest in the image to provide angles data required for orbital information to be derived [58].

### 4.7 Partial Streaks

Partially bound streaks will be generated both entering and exiting the image. Both leading edges, from streaks entering the image, and trailing edges, from streaks exiting the image will be considered. The direction of travel of the RSO will be determined using the predicted position provided by the TLE as described in Section 4.5. The prediction provided will also allow for the determination of residuals by comparing predicted to observed positions.

These residuals can allow for the determination of track error present in the prediction by identifying the difference between the prediction and the observation. It is expected that residuals caused by the TLE will grow as the satellite gets further from the Epoch time provided. Two consecutive nights of data will be considered to determine what role the expected increase in the magnitude of the residuals will have when considering partially bounded and unbounded streaks for the determination of position. Residuals caused by the

---

<sup>11</sup>The right ascension,  $\alpha$ , is defined to be measured positive to the east in the plane of the equator from the direction of the vernal equinox; although it can be reported from  $0^h - 24^h$  in this work degrees,  $0^\circ - 360^\circ$ , are used [8]

<sup>12</sup>The declination,  $\delta$ , is measured northward from the equator ( $0^\circ - 90^\circ$ )[8]

difference between detected pixel locations will also be determined, as discussed below, so that an accurate comparison of the methods can be made.

#### 4.7.1 *Determining Residuals*

To use images in which only single end points of streaks are visible, the end point must be associated with the start or stop time of the image exposure. The challenge is to determine the exact location of the end point: the edge detection method suggested in Reference 65 does not catch the very beginning or end of a streak, as shown in Figure 4.6. Therefore, the pixel identified as the endpoint of the partially bounded streak will have an incorrect time when associated with the image time provided by the meta-data. This lack of correlation results in decreased RSO position accuracy. This effect can occur at both ends when this method is used on bounded streaks.

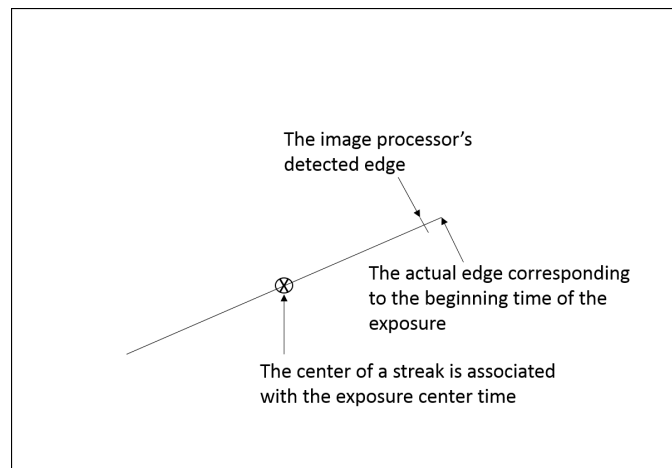


Figure 4.6: The notional difference between a streak's centroid and detected end point

To determine the offset value, a position will be derived using the centroid of a bounded streak. A second position will be found using the detected end point of the partially bounded streak that ends with the same frame used for the bounded frame. Both positions will be compared to the predicted position provided by the TLE. The difference

in the residuals represents the loss of accuracy. If this difference is small, then the use of partially bounded streaks, with a single end point in the image, is a viable method for updating position data in a catalog. The RMSE of the residuals obtained across all of the streaks will be used to determine the relative impact of only using the partially bounded streak.

#### 4.8 Unbounded Streaks

The three nights worth of data on Intelsat 1R were also used to generate unbounded streaks. To determine the usefulness of the position data in a unbounded streak orbital estimation was required, see Appendix C. The control group was used on its own to create orbital data. Unbounded streaks were then substituted for each night in turn to determine the impact of a unbounded streak on the orbital data for each of the first two nights. As shown in Table 4.3, metrics for both the extracted angles and the orbits were recorded for analysis.

Table 4.3: Simulated Data Sets from Collected Imagery of Intelsat 1R

		Night 1	Exposure (sec)	Night 2	Exposure (sec)	Night 3	Exposure (sec)
		# Images		# Images		# Images	
Control Group		8	4-6	8	4-6	8	4-6
Data Set 2	a	0	-	8	4-6	8	4-6
	b	8	4-6	0	-	8	4-6
Data Set 3	a	1	186	8	4-6	8	4-6
	b	8	4-6	1	181	8	4-6

The direction of travel of the RSO (through the image) was determined using the predicted position from the TLE. The prediction also allowed us to determine residuals by comparing predicted to observed positions. The full state covariance of the satellite and the Mahalanobis distance ( $\Psi$ ) were tracked. These metrics were used to assess the usefulness of the unbounded streaks.

#### 4.8.1 Determining Possible Position and Time

Since the RSO is known, the velocity for a circular orbit can easily be determined using Equation (4.4)[87]:

$$v = \sqrt{\frac{\mu_{\oplus}}{r}} \quad (4.4)$$

With the velocity of the satellite determined, the calculation of the distance traveled by the RSO in pixel space for a given exposure length is a multi-step process. The first step is to calculate the distance traveled in kilometers, using Equation (4.5). Then, convert that distance to *radians*, using Equation (4.6), and finally *arc – seconds*, using Equation (4.7).

$$d = vt \quad (4.5)$$

$$\theta = \tan^{-1}\left(\frac{d}{r}\right) \quad (4.6)$$

$$\theta_{asec} = \frac{\theta}{4.8481368111 \times 10^{-6} \frac{rad}{arc-seconds}} \quad (4.7)$$

To convert to pixel space, processing the image through *astrometry.net* returns several calibration variables including the number of *arc – seconds* found in each pixel which is provided in the *pixscale* field. With this knowledge, Equation (4.8) allows for determination of the distance traveled by the RSO in pixel space during the time of the exposure:

$$d_{px} = \frac{\theta_{asec}}{pixscale} \quad (4.8)$$

Once the distance traveled by the RSO has been determined, the location of the motion also needs to be determined. The original simulated image (see Figure 4.4), can help bound the problem. By definition, the unbounded streak touches two edges of the image. This knowledge provides both a No Earlier Than (NET) and No Later Than (NLT) time of

observation. The NET point corresponds to the RSO starting the exposure just inside the image frame and moving the distance  $d_{px}$  in the direction of travel, as shown in Figure 4.7. Determining the time required for the RSO to reach the center of the streak observed in the image provides one possible point in time to bound the RSO's position.

The NLT point corresponds to the exposure ending immediately after the RSO reaches the opposite edge of the frame from the NET point, having moved the distance  $d_{px}$  in the direction of travel, as shown in Figure 4.8. Determining the required time for the RSO to move from the center of the frame to this edge provides the other time bound needed to solve the problem.

Time of the centroid point is important as that will be the single observation used for the orbit update when unbounded streaks are included. The NET and NLT time bounds and the time between them represent possible times that the RSO may have occupied the centroid location to determine the best possible match using a global nearest neighbor approach, as discussed below.

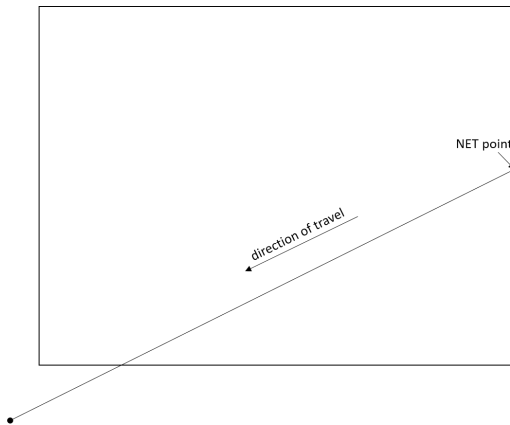


Figure 4.7: A notional NET streak

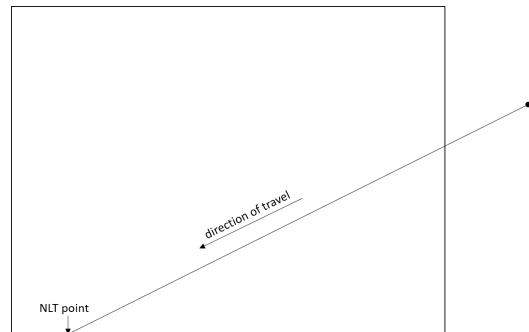


Figure 4.8: A notional NLT streak



## 4.9 Results and Analysis

The results of the experiments presented in Sections 4.3.1 and 4.8 are given below. The data output of each experiment, the analysis of that data, and the determination of the feasibility and utility of performing positional updates for a space object catalog using the metric data obtained from the streaks gathered are provided.

### 4.9.1 *Partial Streaks Results and Analysis*

#### 4.9.1.1 *Position Accuracy when using SDP4*

To determine the standard deviation and the bias of each observation error, an initial calibration was performed. Beginning with a TLE solution one day prior to the first night of observations and the set of bounded observations, the TLE is propagated to the time of each observation. Then, the states are transformed to a predicted observation and the residuals determined for both  $\alpha$  and  $\delta$  of the centroids of the bounded group. After all the residuals are collected, a Gaussian distribution is fit to all the residuals of both observations using a maximum likelihood estimation. The mean of the distribution is used as the bias for the observed data (image). This process determined a bias of  $0.0208^\circ$  for right ascension and  $0.0016^\circ$  for declination, with standard deviations of  $0.0296^\circ$  and  $0.0015^\circ$ , respectively. This bias attempts to account for the constant propagation, sensor, atmospheric, and time error present in the observation leaving only the randomly varying noise and the actual position differences between the propagation and observation present in the residual.

#### 4.9.1.2 *Determined Residuals*

The residuals from the observations of Intelsat 1R are presented in Figures 4.9 and 4.10, showing the difference in radians between the position provided by propagating the TLE and the observed position for both bounded and the partially bounded streak for the same moment in time. The difference between the predicted position and the observed position of the center of the bounded streak is approximately  $517 \mu rad$  in right ascension,

or approximately 22 *km* of along-track error, and 26  $\mu\text{rad}$  in declination, or about 1 *km* cross-track error.

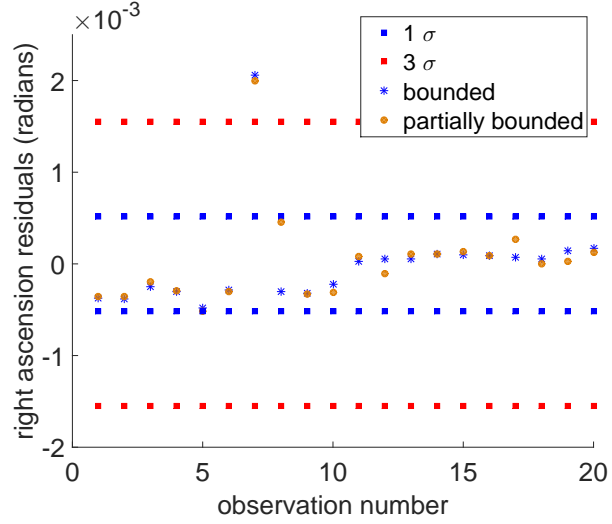


Figure 4.9: The determined right ascension residuals

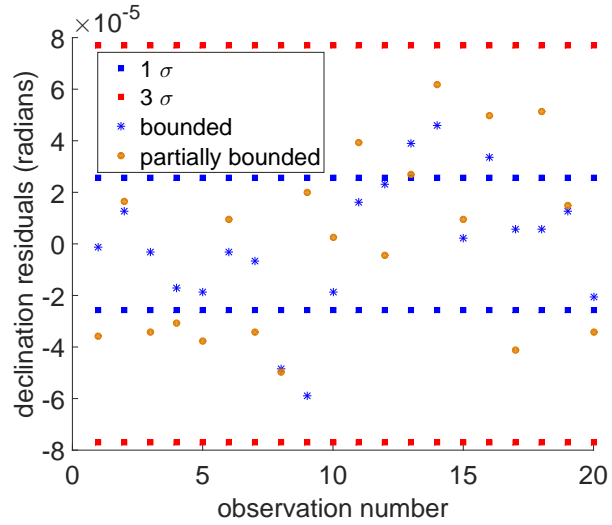


Figure 4.10: The determined declination residuals

Figures 4.11 and 4.12 plots the difference between the determined position of each streak when using a detected end point versus a bounded centroid in the final frame used in the stack. As shown in Figure 4.11 the majority of the difference between the methods is expressed in right ascension, staying below  $200 \mu rad$ , or  $10 km$  difference. The declination difference is below  $90 \mu rad$ , or  $3.8 km$  difference, as shown in Figure 4.12.

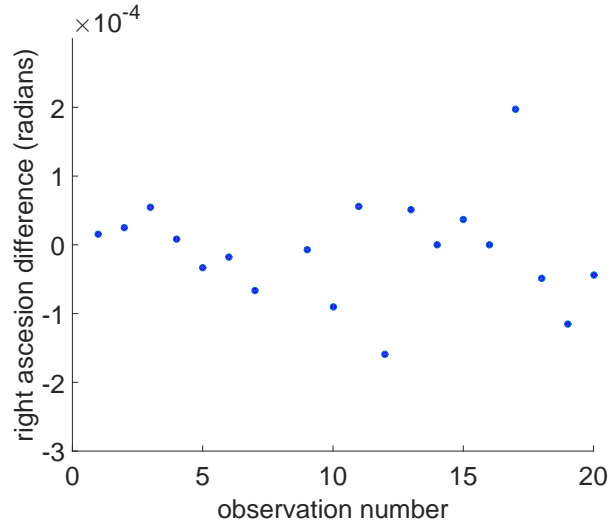


Figure 4.11: The right ascension difference between traditional and experimental methods

Overall an RMSE analysis of the residuals, provided in Table 4.4, shows an offset less than  $3 \mu rad$  in right ascension and  $12 \mu rad$  in declination is apparent between the use of the center of the streak and a detected end point. When compared to the predicted offset provided above; both the right ascension and the declination observed error actually perform better than expected, but are still in line with past studies [43, 44].

#### 4.9.1.3 End Point Localization

These results presented so far confirm that the prediction model is working as expected and that the difference between bounded and partially bounded streaks is minimal. To

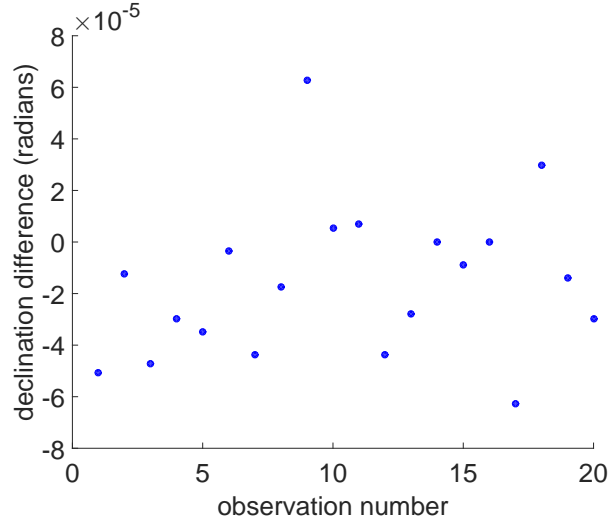


Figure 4.12: The declination difference between traditional and experimental methods

Table 4.4: RMSE for positional data

Type	Right Ascension ( $\mu rad$ )	Declination ( $\mu rad$ )	Position Difference Along-Track ( $km$ )	Position Difference Cross-Track ( $km$ )
bounded	516.52	25.70	21.78	1.08
partially bounded	513.80	34.35	21.66	1.45
Difference	2.72	11.65	0.12	0.37

ensure that using the endpoint of the streak over the centroid of a bounded streak is not unnecessarily introducing new error requires further analysis. To ensure the algorithm is accurately localizing the end of the streak, a comparison was made between the detected endpoint, in pixel space, and the single frame with the same associated time. The residuals determined between the two methods is provided in Figures 4.13 and 4.14.

The end point detection tends to identifying a pixel near the edge and end of the streak, and not the exact end point. This offset is caused by the algorithm ignoring light below 80% peak intensity of the streak. To better determine the accuracy of the end

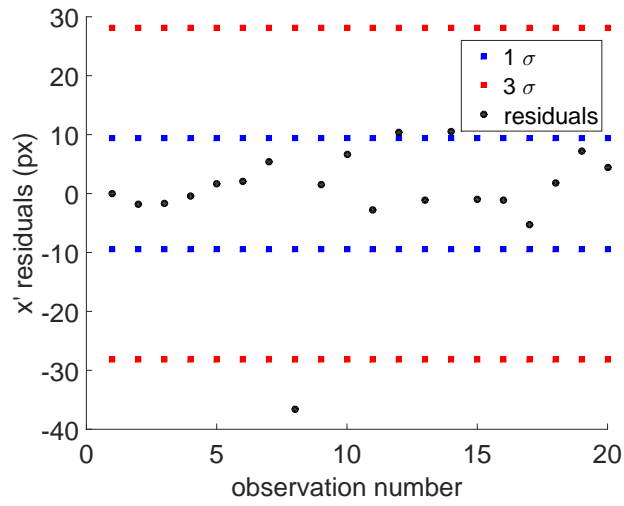


Figure 4.13: The difference in  $x'$  pixels between edge and centroid location

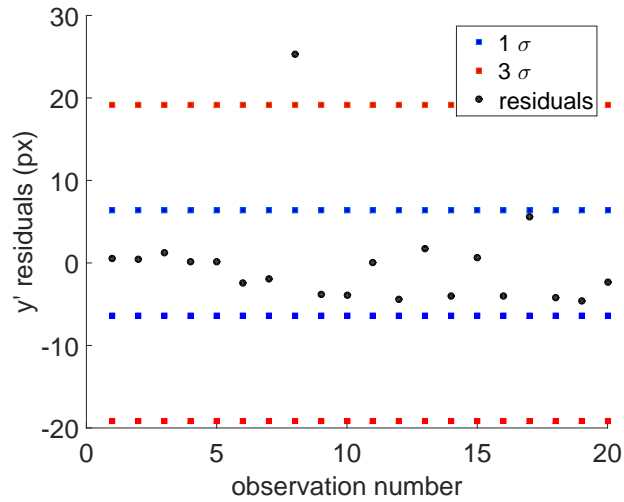


Figure 4.14: The difference in  $y'$  pixels between edge and centroid location

point localization, a Gaussian distribution is fit to all the pixel residuals using a maximum likelihood estimation. This process determined a bias of  $-0.47 \text{ px}$  for  $x'$  and  $1.57 \text{ px}$  for  $y'$ , with standard deviations of  $9.39 \text{ px}$  and  $6.38 \text{ px}$ , respectively. This bias is caused

by the accuracy of the end point detection algorithm compared to the centroid algorithm used. The weighted centroid algorithm used on the individual frames selects the peak intensity creating the difference in selected pixel. This difference is independent from TLE propagation and introduces the 1.43 *km* along-track and 0.6 *km* cross-track offset observed above. As Figures 4.13 and 4.14 show, with the exception of a single outlier (observation 8), all of the pixels selected by the end point detection algorithm fall within a three standard deviations, and most fall within a single standard deviation of the location of the centroid provided by the single frame measurements. Observation eight in Figures 4.13 and 4.14 occur when the RSO is near a bright star, and the proximity to the star clearly impacts the algorithm determination of the endpoint.

#### ***4.9.1.4 Impact to Current Operations***

Although telescope operators state that streaks happen all of the time, exact quantification of the frequency of streaks is difficult to produce [25, 65]. A quick survey of operating GEO satellites was performed to estimate how often SSA telescopes might observe a serendipitous satellite. Of the 91 operational GEO satellites in the western hemisphere, 30 are within half a degree of another, 43 have less than one degree of separation, and 60 are within three degrees of another vehicle. Therefore, for a telescope with a single degree FOV centered on a GEO vehicle, there is a 33% chance that the sensor will serendipitously capture a second satellite. If a GEODSS sensor, with two degrees FOV was used, it will have a 47% chance of serendipitously capturing a second satellite. If a wide-area sensor, such as Pan-STARRS, was used then there would be a 66% chance of capturing another satellite in the image as a partial streak removing the need to schedule an additional observation [30, 88].

#### ***4.9.2 Unbounded Streaks***

To implement the UKF, a starting covariance is needed to accompany the TLE of interest. Since the candidate RSO is in a GEO orbit and observed frequently, a high

accuracy covariance as proposed by Horwood is used [73]. Define the starting covariance in equinoctial elements such that  $\mathbf{P}_0 = \mathbf{A}^T \mathbf{A}$ , where

$$\mathbf{A} = \text{diag}(0.05 \text{ km}, 10^{-5}, 10^{-5}, 10^{-5}, 10^{-5}, 97 \times 10^{-6} \text{ rad}) \quad (4.9)$$

The initial state is pulled directly from the published TLE.

For the following analysis, the comparison of orbital data generated using three subsets of the data was required. The first set of data is created strictly from the control group, to provide a baseline analysis. The second data set generates an orbit by excluding data from the night with the simulated unbounded streak. The second data set provide estimated orbital positions assuming that the captured unbounded streak was thrown out, and the orbit was determined using only the two nights of data. The final data set incorporates the unbounded streak data. These data sets allow for analysis of the usefulness of the position data in the unbounded streak. The impact of using the unbounded streak vs. ignoring the streak and using fewer observations is discussed below.

#### **4.9.2.1 Position Accuracy**

The unbounded streak experiment used an HPOP propagator and UKF to aid in orbital position determination, so a second calibration was required to accurately account for these differences. The Kalman filter is the optimal estimator when the observation error (noise) is Gaussian and is perfectly captured by the observation noise covariance matrix,  $\mathbf{R}_{\text{noise}}$ . To determine the standard deviation and the bias of each observation error, an initial calibration was performed. Beginning with a TLE solution one day prior to the first night of observations and a set of control data telescope observations, the TLE is converted to equinoctial elements and then propagated to the time of each observation. Then, the states are transformed to a predicted observation and the residuals determined for both  $\alpha$  and  $\delta$  of the centroids of the control group. After all the residuals are collected, a Gaussian distribution is fit to all the residuals of both observations using a maximum likelihood estimation. The mean of the distribution is used as the bias for the observed data (image)

and the standard deviation of the distribution is incorporated in the  $\mathbf{R}_{\text{noise}}$  matrix. This process determined a bias of  $-0.0263^\circ$  for right ascension and  $0.0172^\circ$  for declination, with standard deviations of  $0.0124^\circ$  and  $0.0014^\circ$ , respectively.

#### 4.9.2.2 *Best Streak Observation*

When processing unbounded streak angle data, it is necessary to associate a time stamp with the observation angles. This problem is solved using a discrete optimization search routine, over the range of possible times, that uses the UKF propagation step to determine a metric based on the weighted residual for all possible time steps. The weighted residual metric is defined as

$$J = \left\| \frac{\mathbf{v}_\alpha}{\sigma_\alpha} \right\|_1 + \left\| \frac{\mathbf{v}_\delta}{\sigma_\delta} \right\|_1 \quad (4.10)$$

The best estimate for the time step is selected as the time step associated with the smallest value of  $J$ . Figures 4.15 through 4.17 demonstrate this process for the first night's unbounded streak. As the figures show the best possible time is associated with the line segment 243 which corresponds to 08:27:07 Greenwich mean time (GMT). With the best times associated with the unbounded streaks, an orbital estimation can be performed and an analysis of that orbit conducted.

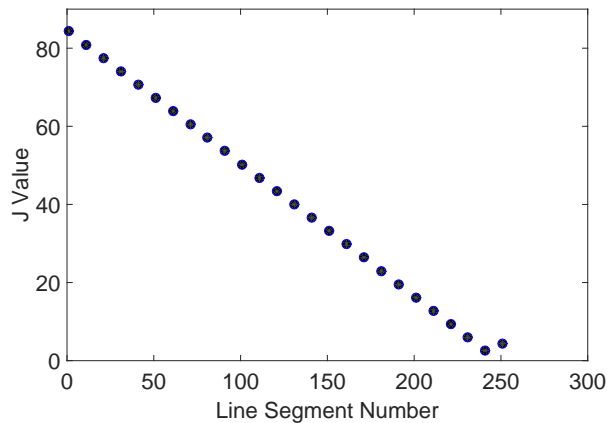


Figure 4.15: First pass through the optimizer



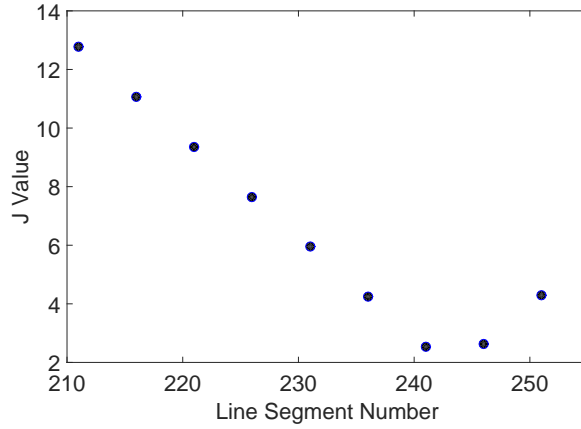


Figure 4.16:  $n^{th}$  pass through the optimizer

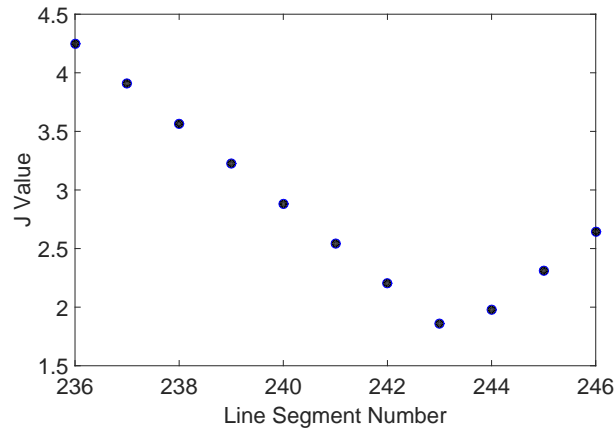


Figure 4.17: Final pass through the optimizer

#### 4.9.2.3 Residuals Analysis

The analysis of the estimated orbits begins with the comparison of the estimated residuals. Figures 4.18 through 4.21 compare the  $\alpha$  and  $\delta$  residuals from the control group to the orbital estimation created while ignoring the night with a simulated unbounded streak (streak excluded), and the estimation using the unbounded streak (streak included) from the first two nights of observation. Analysis of Figures 4.18 through 4.21 shows very little

difference between these residuals with a RMSE difference of  $0.00025^\circ$  in right ascension, and  $0.00093^\circ$  in declination.

The largest differences are observed in declination on the second night of observations when the unbounded streak is observed on the first night, see Figure 4.19; however, once the second night of observations are included the presence of the unbounded streak improves the solution. This improvement is evident in the third nights residuals, observations 10 forward in Figure 4.19. Observation nine in Figure 4.19, which is also observation one in Figure 4.21, is greater than three sigma from our expected observation. This observation represents the earliest frame with the RSO present on the third night and due to both increased atmospheric turbulence, and the proximity to the edge of the sensor's FOV suffers from increased distortion. Because this increased distortion is present in all three data sets, including the outlier does not impact the results.

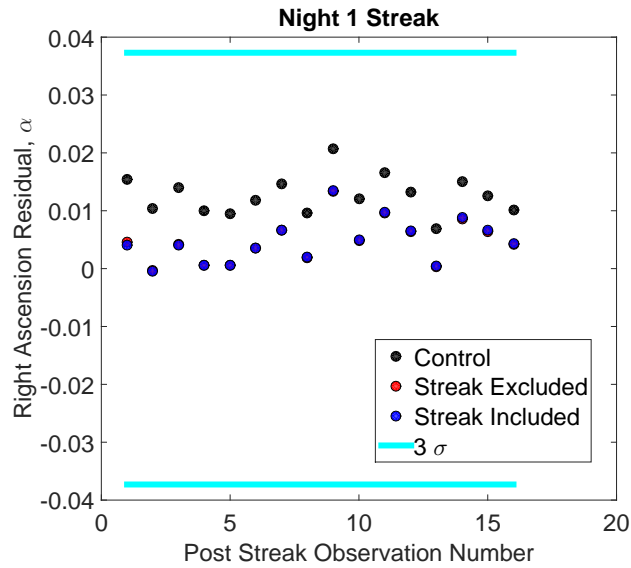


Figure 4.18: Estimated  $\alpha$  residuals of 2<sup>nd</sup> and 3<sup>rd</sup> night (1<sup>st</sup> night unbound streak)

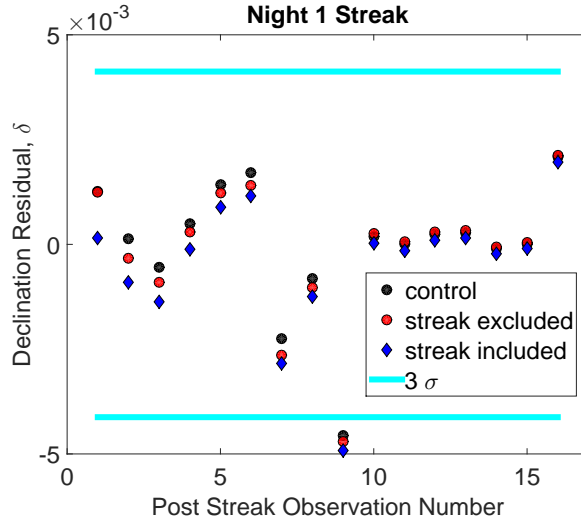


Figure 4.19: Estimated  $\delta$  residuals of 2<sup>nd</sup> and 3<sup>rd</sup> night (1<sup>st</sup> night unbound streak)

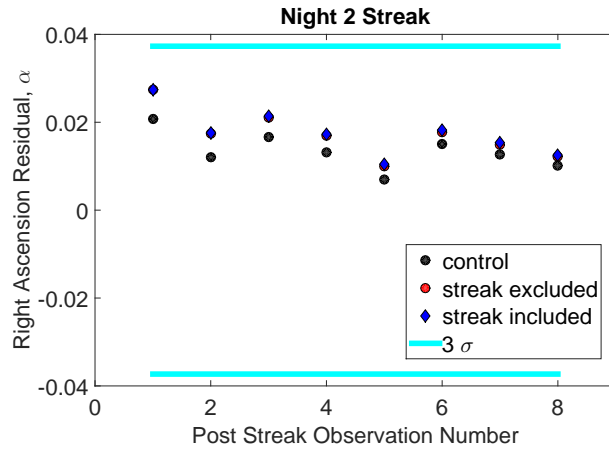


Figure 4.20: Estimated  $\alpha$  residuals of 3<sup>rd</sup> night (2<sup>nd</sup> night unbound streak)

#### 4.9.2.4 Covariance Analysis

Figures 4.22 through 4.25 compare the covariance in mean longitude and semi-major axis using observations from the control group and the simulated unbounded streaks, both estimations ignoring the night with a simulated unbounded streak (streak excluded), and the estimation using the unbounded streak from the first two nights of observation (streak

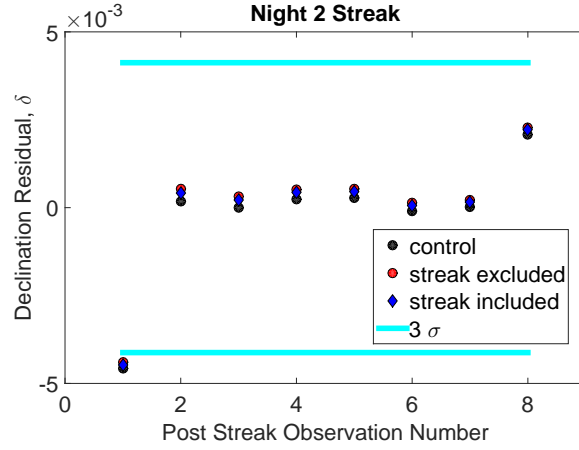


Figure 4.21: Estimated  $\delta$  residuals of  $3^{rd}$  night ( $2^{nd}$  night unbound streak

included). These orbital elements were chosen because the orbital position in physical space is well-represented by these elements.

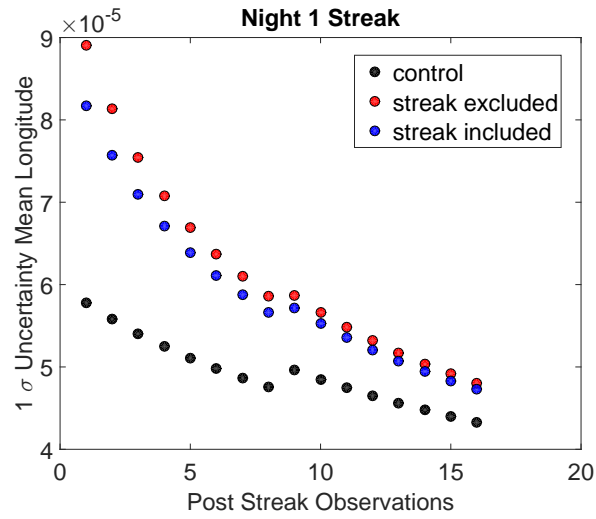


Figure 4.22: Mean longitude covariance

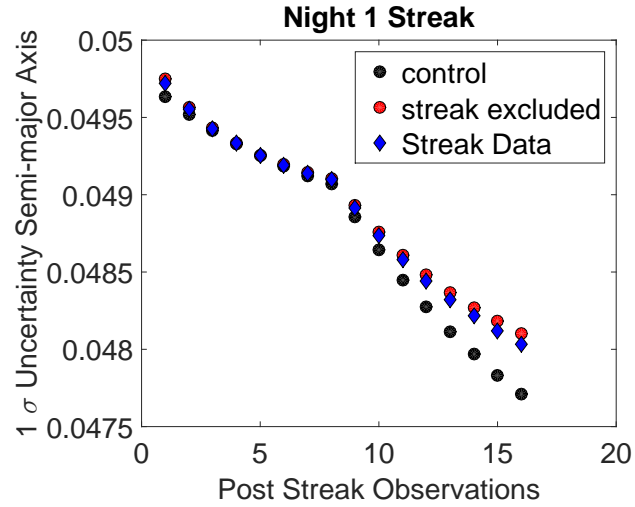


Figure 4.23: Semi-major axis covariance

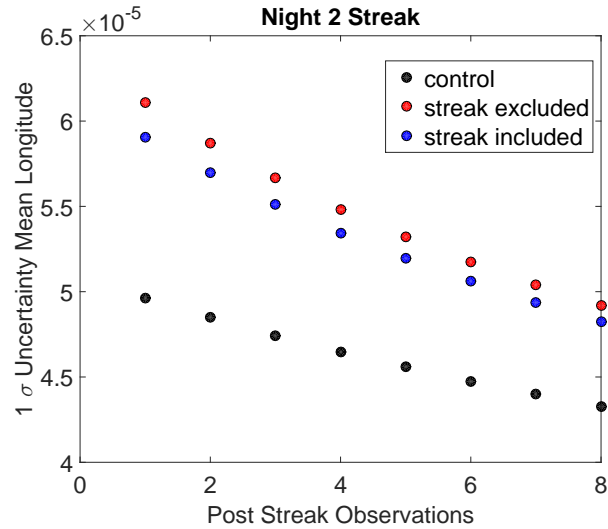


Figure 4.24: Mean longitude covariance

In all cases, using the unbounded streak data drives the covariance closer to the control group. As the measurement errors have been determined this improvement is expected; any observation, regardless of quality, should drive down the covariance, although the improvement doesn't necessarily converge to the true position. Therefore, the Mahalanobis

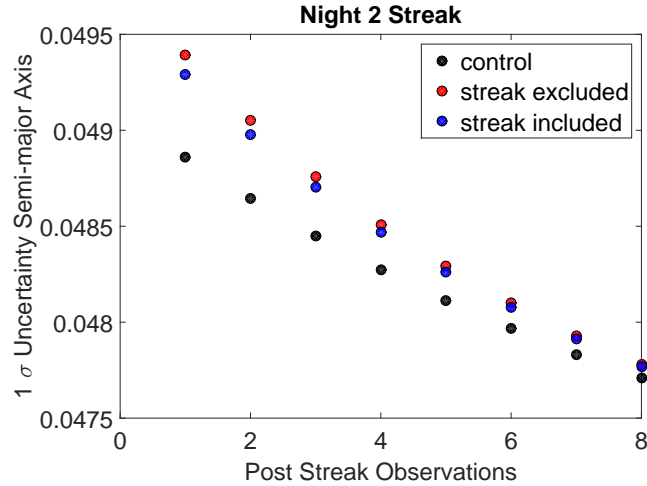


Figure 4.25: Semi-major axis covariance

distance is required to determine if this decrease in the size of the covariance is actually an improvement.

#### 4.9.2.5 Mahalanobis Distance Analysis

Figures 4.26 and 4.27 show the Mahalanobis distance using observations from the control group along with the orbital estimation, which did not include the night with a simulated unbounded streak (streak excluded) and the estimation using the unbounded streak data (streak included) from the first two nights of observation. Although the data does show that the use of the unbounded streak data can be slightly less accurate (with an RMSE value of 0.2), 80% of the time the Mahalanobis distance confirms that the orbital fit using the unbounded streak data is equal to or better than the resulting fit when ignoring the unbounded streak data. For more information on Mahalanobis distances and how it was calculated here see Appendix C

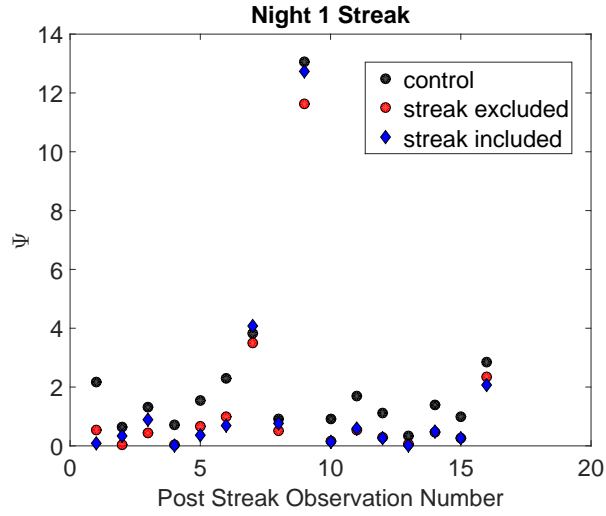


Figure 4.26: Night 1 Mahalanobis distance

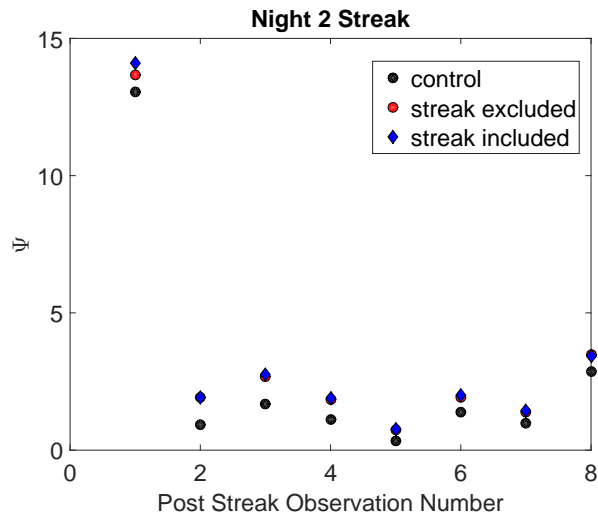


Figure 4.27: Night 2 Mahalanobis distance

## 4.10 Summary

The experiments performed demonstrate that for the available real-world data used here partially bounded and unbounded streaks can provide useful orbital position data for RSOs in astronomical imagery. End point localization, provided in Section 4.9.1.3, allows

for partially bounded streaks to be used. These partially bound streaks provide a mean degradation of 3.88% in right ascension and 259.41% in declination, when compared to the control group.

The application of a global nearest neighbor time estimate, provided in Section 4.9.2.2, for the unbound streaks allows for the reconstruction of the RSO's orbit that 80% of the time is equal to or better than the orbit reconstructed while ignoring the unbound streak, as shown in Sections 4.9.2.3 through 4.9.2.5. The impact of including unbound streaks provides a mean degradation of 6.2% in right ascension and 42.69% in declination, when compared to excluding the unbound streaks when reconstructing the orbit.

In both experiments the percentage degradation of the cross-track position is worse than the along-track degradation for two reasons. First, the cross-track position error is smaller, exaggerating any difference. Second, the edge detector used for pixel localization picks a pixel on the edge of the streak not at its center.



## **5. Conclusions and Future Work**

The work here in sought to enable the use of external sensors for SSA as discussed in Chapter 1, by building upon past work in extracting streaks from astronomical images as found in References 61, 65, 89, and 90. By incorporating Levesque's work in Reference 65, in series with Lang's work in Reference 58, and orbital determination techniques found in References 8 and 91, partially bound and unbound streaks were used to accurately reconstruct an orbit of a known RSO using real-world data. The findings support the operational utility of more sensors, and may enable secondary SSA missions from current AFRL sensors [5, 25, 92]. This chapter will provide a breakdown of the conclusions of the two phases of research, the answers to the proposed research questions, and a recommendation for future work to build upon the findings presented.

### **5.1 Phase I Conclusions**

The first phase, detailed in Chapter 3, sought to find the best data calibration technique for extracting time, right ascension, and declination data from images. The best calibration technique was found by comparing several existing techniques to determine the sensitivity of each process to the distortions found in EO data. The focus of Phase I work was to answer the following research questions:

1. What is the level of accuracy achieved by each calibration method?
2. Which method best accounts for image distortion?

The analysis provided in Chapter 3 indicates an uncalibrated image including a streak created by an RSO can be processed to extract angles data. All three astrometry methods explored return simulated position data to within  $0.2^\circ$  of the true position without prior calibration. These results are representative of potential serendipitous SSA sources, as most

astronomical data sources will provide images that are well defined using a star catalog. Other astrometry methods may also provide equally accurate results, and depending on the source of the noise, a given method may be preferred over the others.

The results of this phase were compared to Sease et al. using *astrometry.net*, to perform the astrometry process in *GEODETICA* in Reference 59, published in early 2015. Lang provides a complete description of the *astrometry.net* geometric calibration in Reference 58 and it outperforms the intrinsic parameters calibration for real world images, by approximately  $0.5^\circ$ , therefore *astrometry.net* is the recommended tool for processing real world images that include enough stars for a successful calibration.

The RMSE analysis results, provided in Figure 3.11, shows that performing an intrinsic parameters calibration outperforms the remaining methods by a range of  $0.04 - 0.22^\circ$  depending on the noise source, clearly demonstrating that it is preferred over the other two methods for serendipitous streak harvesting if *astrometry.net* cannot be used. The requirements for knowledge of the intrinsic parameters might limit the intrinsic parameters methods use to known systems, but the real world results indicate that even without the intrinsic parameters the pinhole camera model returns an almost equally accurate position for diffraction limited systems. The findings of this phase clearly show that accurate orbital reconstruction can be performed using an RSO streak in a distorted image, without applying calibration frames (dark, flats, and bias frames).

## 5.2 Phase II Conclusions

The work of Phase II, detailed in Chapter 4, examines the feasibility and utility of performing in-track, cross-track, and orbital element updates for an RSO catalog using metric data obtained from streaks gathered by astronomical telescopes. This phase builds on the findings of Phase I by using the astrometry methods to determine accurate angle data and reconstruct orbital position. The focus of this work was on gathering and using three possible data categories: streaks that only enter, only exit, or cross completely through

the astronomical image. The focus of Phase II work was to answer the following research questions:

1. Can observations of tracks only partially in the image be used to improve SSA?
2. Can observations of tracks passing fully through the image be used to improve SSA?

The data used for this phase was gathered using low cost Commercial Off the Shelf (COTS) equipment, located in data Ohio. The nature of the equipment and its location provides real-world data that is less than ideal and represents the lower end of quality that would be expected from astronomical scopes.

#### ***5.2.1 Using Partially Bounded Streaks***

As shown in Section 4.9.1, high confidence exists that the method used correctly isolates the end of a streak and selects a pixel associated with the position of the observed RSO. Although the pixel selected is offset by the threshold chosen to indicate the end of the streak and not the actual pixel location that correlates to exposure's time stamp, it is shown to be within a single standard deviation.

These partially bound streaks provide a mean degradation of 3.88% in right ascension and 259.41% in declination respectively when compared to the control group. This difference is reflected in the  $3 - 12 \mu rad$  offset found when using streaks with a single end point versus using a streak's centroid. The impact of using a single end point corresponds to a difference of less than 0.4 km of orbital positional discrepancy for objects in GEO orbits. These differences can be directly attributed to differences in the detected pixels used to determine the angles used in the orbit reconstruction. Further, the difference is significantly smaller than the detected position difference of more than 22 km caused by propagation inaccuracies.

These results support the idea that using a partially bounded streak, can be used for a satellite position update, as the observation provides a more accurate location for the

RSO than the propagated TLE alone, even with the offset caused by the experimental method. The decrease in track error provided exceeds the small offset the new method creates, allowing for enhanced SSA from the use of partial streaks when full streaks from dedicated RSO observations are unavailable.

### ***5.2.2 Using Unbounded Streaks***

The combination of the decreasing covariance and the equally good Mahalanobis distance confirm that unbounded streaks can provide useful position and velocity information. The covariance could be improved further by using data with sub-second time accuracy, or gathered at a location with less atmospheric turbulence, which would increase the resolution of the data. The impact of including unbound streaks provides a mean degradation of 6.2% in right ascension and 42.69% in declination, when compared to excluding the unbound streaks when reconstructing the orbit. These results support the idea that using a unbounded streak can provide position and velocity data, allowing enhanced SSA from the use of unbounded streaks in conjunction with dedicated RSO observations to perform an orbital update.

In addition to the potential efficiencies in tasking GEODSS sensors, discussed in Section 4.9.1.4, the ability to use both partially bound and unbound RSO streaks may enable existing sensors to contribute to the SSA mission. For example AFRL has a number of cold staring optical sensor that observe these types of streaks, and it is expected that LSST will also observe multiple satellites each night it operates [25, 78]. Both of these resources might provide valuable data, but because this study was limited in scope additional work needs to be performed to further study the suitability of serendipitous satellite streaks.

## **5.3 Future Work**

The data used for this work was a single satellite (Intelsat 1R) observed by a single optical system over a three night period and processed by a time consuming method that

still requires significant manual input. Follow-on work should be focused on one of three areas: automation, serendipitous collects, and handling several observations from multiple sensors.

True automation of the image processing method outlined in Appendix A would allow real time angle extraction from these images enhancing SSA. Currently the threshold, minimum streak length, the size of the gap (caused by staking) to ignore, and the number of peaks to include all have to be set manually, see Section A.1.1.2, if these values could be automatically detected each image could be processed in less than a minute. Alternatively, *GEODETICA* provides exactly the type of automation required, but it is not yet able to detect end points of partially bounded streaks, or process unbounded streaks. Adding this capability to *GEODETICA* would make it ideal for processing serendipitous streaks.

Although the experiments conducted in this work indicate that astronomical sources can be of use to global SSA, the data was provided by AFIT's TeleTrak system. The data was collected while observing young stellar objects over a three night period, but future work should attempt to get real world astronomy data, or AFRL serendipitous collects for processing.

Future work should focus on confirming that these methods also work for other orbital regimes. Streaks serendipitously captured by optical systems have a high probability of being Low Earth Orbit RSOs. The TLE for Low Earth Orbit (LEO) objects are considered less accurate than they are for the GEO objects studied here. The same experimental methods used here can be applied to bright LEO objects imaged with rapid exposure times once LEO data has been made available.

Serendipitous streak data from multiple locations provides the best opportunity to leverage external sensors and maximizes their impact on SSA operations. The increase in awareness provided by the external sensors, as well as the benefit of geographic separation needs to be balanced against data fusion challenges. Exploring the correct methods for

integrating this data in a timely manner will be required before external sensor data can be used operationally.

## Appendix A: Image Processing Methods

Each step in the process of extracting angles from an RSO streak used in this dissertation are well defined in other sources, see References [8, 33, 56, 58, 90], but are combined here to create a functioning angle extraction tool. This appendix will cover how the images in the data set were processed, including a brief overview of the code used. The overall process is provided in Figure A.1 with each step covered in the sections below. The images provided throughout this appendix are examples taken from the data set described in Chapter 4.

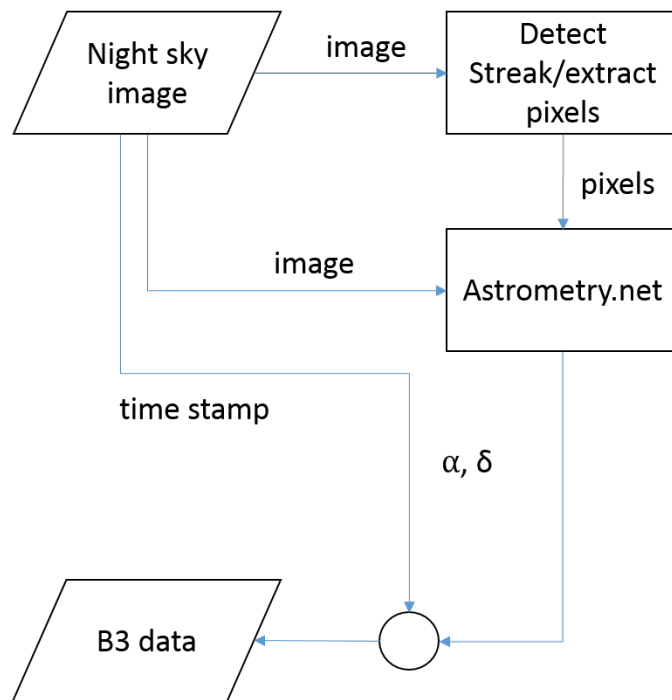


Figure A.1: Image Processing Method

## A.1 Data Reduction

To reduce the data provided by TeleTrak the images can be converted to pixel locations using the script provided in Listing A.1

Listing A.1: Script used to reduce images to pixel coordinates

---

```
1 %Used to reduce the images in DataAstrometryPaper to the required pixel
2 %locations.
3 close all; clc;
4
5 %% Create the directory of images to process
6 % Point the processor at a file of png images
7 fileFolder='C:\Users\Charlie\Desktop\PartialStreaks\25Sept14';
8 FileZ=dir(fullfile(fileFolder,'*.fits'));
9 %assume a sperical earth for now, this will be more accurate if you use
10 %WGS data to perfect the radius to the scope location.
11 r_earth=6378.135;
12 for ii=2:length(FileZ);
13 %Process each image returned to extract the streak coordinates
14     Image=fitsread(fullfile(fileFolder,FileZ(ii).name));
15     xy_long(:,ii) = image_processing_d( Image );
16 end
```

---

The script given in listing A.1 performs the data reduction by calling two functions. Lines 7 and 8 define a directory of images to be processed. Lines 9-15 loops through the images in the directory to reduce them to pixel locations. Specifically, line 15 processes each image using the function described in Section A.1.1.

### A.1.1 Detect Streak and Extract Pixel

In Listing A.1 line 12 passes each image to the *image\_processing* function provided in Listing A.2. This function uses several built in *MATLAB* functions to process the image to extract the required pixel locations for use in the camera model provided in Section 2.4.3. Figure A.2 is an example image from the data set and will be processed in this appendix.

#### A.1.1.1 Detecting RSO streaks

The steps taken in Listing A.2 are a subset of the steps provided by Levesque. Primarily, first the background was modeled and removed, next the bright stars identified and removed, followed by a Hugh transform performed as described in Reference 65. Applying Levesque's method would decrease the required visual magnitude of an RSO



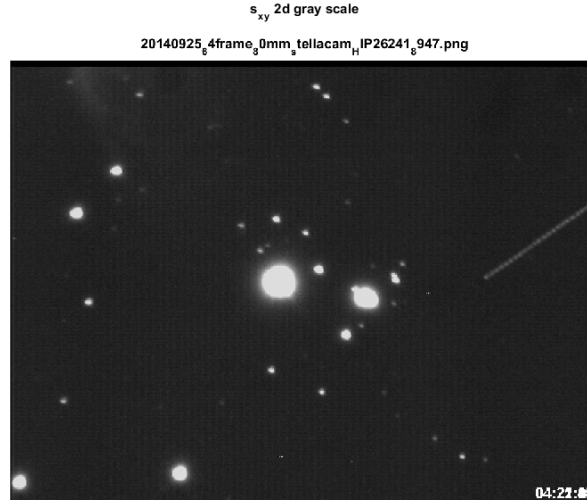


Figure A.2: Image ( $S_{xy}$ ) Returned by the System

for detection, but was not required for the streaks in the data used here. Listing A.2 is a function that requires an image and returns the pixel locations of detected streaks in the *xy\_prime* matrix and calls out the pixels of the longest streak in *xy\_long*.

Listing A.2: Process an image to identify RSO streaks

---

```

1 function [ xy_long, xy_prime ] = image_processing( Image )
2 %Function created by: C. Bellows, AFIT 2014
3 %Process image to remove background, stars, detect streaks and extract
4 %pixel locations using a modification of Levesque et. al. method published
5 %in 2007. (cleaned up version!)
6 %This function takes the processed streak images of streaks and processes them using an edge
7 %detector to return an array of end point pixels for each streak in the
8 %image.
9 % Inputs:
10 % Image      the rotated image s_xy
11 % Outputs:
12 % xy_prime   the xy pixel locations of the end points of the streaks
13
14 %%step 1: preprocess the image to enable analysis
15 %need to convert to uint8 as matlab struggles with fits file manipulation.
16 I1=mat2gray(Image);
17 %need to convert to a 2D matrix:
18 d=size(I1);
19 J=zeros(d(1,1),d(1,2));
20 J(:,:)=I1(:, :, 1);

```

```

21 figure(23);imshow(J);title('s_x_y 2d gray scale');
22
23 %step 2: Model and remove image background
24 %remove all foreground using morphological opening. The opening operation
25 %removes objects that cannot completely contain the structuring element.
26 background=imopen(J,strel('disk',35));
27 %view the background approximation as a surface to see where illumination
28 %varies.
29 figure(24);
30 surf(double(background(1:8:end,1:8:end))),zlim([0 255]);
31 set(gca,'ydir','reverse');
32 %subtract the background approximation image, from the original image, the
33 %resulting image has a uniform background.
34 J2=J-background;
35 figure(25); imshow(J2); title('s_x_y background removed');
36
37 %step 3: adjust the contrast by 1% at both the high and low ends of the
38 %spectrum.
39 J3=imadjust(J2);
40 J3=imsharpen(J3);
41 figure(26);imshow(J3);title('s_x_y contrast enhanced');
42
43 %%step 4: Remove stars:
44 %image segmentation
45 level = graythresh(J3);
46 BW = im2bw(J3, level);
47 image=~BW;
48 se = strel('disk',6,6);
49 image_a = imerode(image,se);
50 image2=J3(:,:,1);
51 image3=image2.*image_a;
52 image4=imsharpen(image3, 'radius', 10);
53 figure(27); imshow(image4); title('s_x_y star masked');
54
55 %%step 5: change to binary and pass to streak extractor_2
56 s_xy=image4>level-.2;
57 figure(28); imshow(s_xy); title('s_x_y streak');
58
59 %step 6: perform the hugh transform and return the x' y' pixel location of the
60 %streaks.
61 [ xy_long, xy_prime ] = streak_extraction_2( s_xy, Image );
62 end

```

---

In Listing A.3 lines 14 through 21 convert the image to a two dimensional matrix. First, line 16 converted the image into a gray-scale image. Second, line 18 determines the height and width of the image. Then, line 19 creates a matrix of the correct size, and line 20 fills that matrix with the correct numbers. Finally, line 21 displays the matrix as an image, as shown in Figure A.2.

As with Levesque's method the next step is to model and remove the background of the image. This removal is to is accomplished from Steps 23-35. Line 26 removes the foreground using morphological opening. The opening operation removing the objects

that *MATLAB* cannot completely contain the structuring element. Lines 29-31 displays the background approximation, given in Figure A.3. Line 34 of Listing A.3 then subtracts the background surface from Figure A.2 and line 35 displays the background free image, given in Figure A.4.

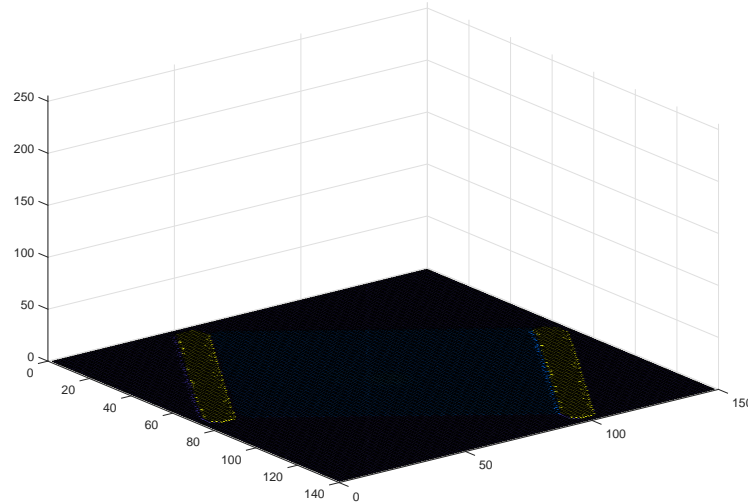


Figure A.3: Mapped image background

To sharpen the edges of objects in the image lines 37-41 adjust the contrast of the image by 1% at both the high and low ends of the spectrum. Line 39 adjusts the low end of the spectrum, using *imadjust*. Then, line 40 adjusts the high end of the spectrum using *imsharpen*. The results are displayed using line 41 as shown in Figure A.5.

Once the contrast has been enhanced removing the stars is accomplished with lines 43-53. Line 45 determines the threshold separating the foreground objects from background noise. Line 46 then converts the image to black and white using this threshold level. Line

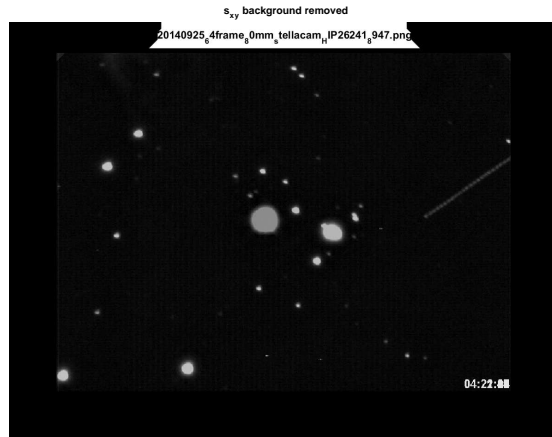


Figure A.4: Image after mapped background removed

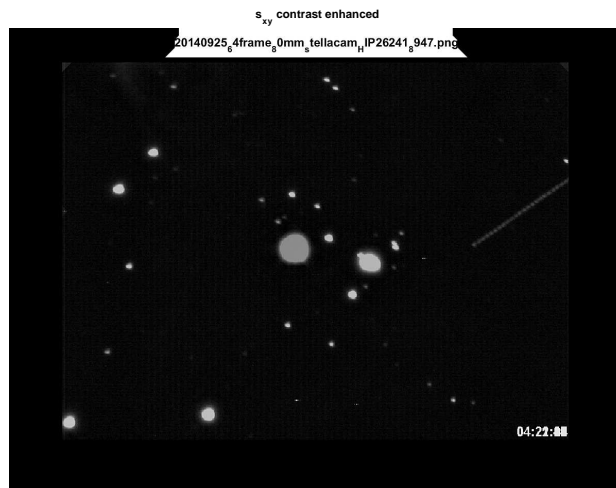


Figure A.5: The contrast enhanced image

48 creates the required strel disk and line 49-51 places the disk over each star. Line 52 sharpens the remaining image for display on line 53, as shown in Figure A.6.



Figure A.6: The star masked image

Lines 55-57 converts the image to binary. Line 56 performs the conversion using a threshold slightly below the threshold automatically detected by *MATLAB*, to ensure the streak remains visible. Line 57 displays the new image, provided in Figure A.7. Once a binary image of the streak is available it is passed to the streak extraction function discussed below, on line 61, which returns the pixel locations of the detected endpoints of all streaks as well as marking the longest streak detected <sup>13</sup>.

#### ***A.1.1.2 Determine Pixel Locations***

Once the binary image has been passed to the streak extraction function, given in Listing A.3, the edges can be detected and a Hugh transform performed to extract pixel locations for the streak. Listing A.3 is a function, called by Listing A.2 that requires a binary image (*s\_xy*) and the original fits image, and returns the pixel locations of detected streaks in the *xy\_prime* matrix and calls out the pixels of the longest streak in *xy\_long*. The

---

<sup>13</sup>for the experiments described in Chapter 4 these endpoints where used to determine the centroid of the streak, except when the end points where required for the experiment, as described in Section 4.7

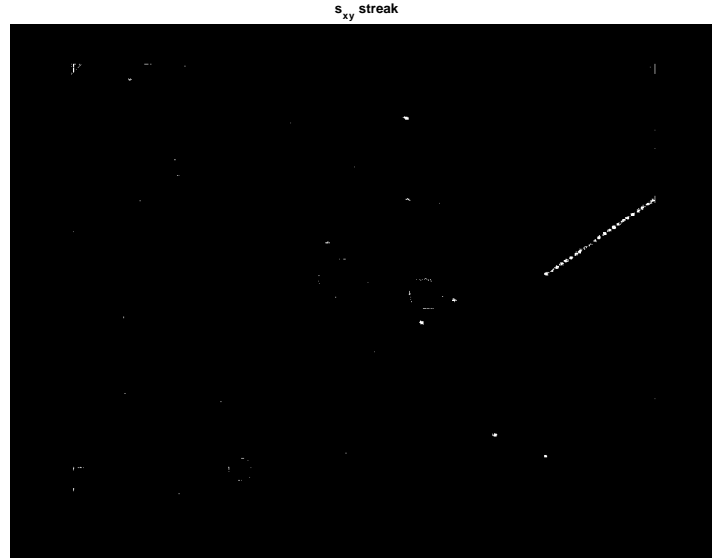


Figure A.7: The binary streak image

function provided in Listing A.3 also marks the streaks on the original image, the longest streak is colored red, and any secondary streaks would appear green.

Listing A.3: Finding the pixel location of image streaks

---

```

1 function [ xy_long, xy_prime ] = streak_extraction( s_xy, I1 )
2 %streak_extraction returns pixel locations for streaks in an image
3 %Function created by: C. Bellows, AFIT 2014
4 % This function takes the processed streak image and processes them using
5 %a Hugh transform to return an array of end point pixels for each streak in
6 %the image.
7 % Inputs:
8 %   s_xy      the image file created by image_processing_c.m
9 %   I1        the fits file returned by astrometry.net
10 % Outputs:
11 %   xy_prime  the xy pixel locations of the end points of the streak
12
13 %% Satellite streak detection
14 %% -----
15 %Extract edges.
16 BW = edge(s_xy,'canny'); figure(24); imshow(BW);
17
18 %Display the Hough matrix.
19 [H,theta,rho] = hough(BW);
20 figure(25), imshow(imadjust(mat2gray(H)),[], 'XData',theta,'YData',rho,...
21 'InitialMagnification','fit');
22 xlabel('\theta (degrees)'), ylabel('\rho');
23 axis on, axis normal, hold on;

```

```

24 colormap(hot)
25
26 %Find the peaks in the Hough transform matrix, H, using the houghpeaks
27 %function.
28 P = houghpeaks(H,50,'threshold',floor(0.1*max(H(:)))));
29 %Superimpose a plot on the image of the transform that identifies the peaks
30 xy_prime = theta(P(:,2));
31 y = rho(P(:,1));
32 plot(xy_prime,y,'s','color','black');
33 %Find lines in the image using the houghlines function.
34 lines = houghlines(BW,theta,rho,P,'FillGap',20,'MinLength',500);
35
36 %Create a plot that superimposes the lines on the original image.
37 d=size(lines);
38 temp=zeros(2,2,d(1,2));
39 figure, imshow(I1), hold on, title('Detected Streaks');
40 max_len = 0;
41 for k = 1:length(lines)
42     xy = [lines(k).point1; lines(k).point2];
43     temp(:,:,k) = [lines(k).point1; lines(k).point2];
44     plot(xy(:,1),xy(:,2),'LineWidth',2,'Color','green');
45     % Determine the endpoints of the longest line segment
46     len = norm(lines(k).point1 - lines(k).point2);
47     if ( len > max_len)
48         max_len = len;
49         xy_long = xy;
50     end
51 end
52 % highlight the longest line segment
53 plot(xy_long(:,1),xy_long(:,2),'LineWidth',2,'Color','red'); hold off;
54
55 %% get the pixel values of interest
56 [ix,iy] = find(temp > 0);
57 c = [ix iy];
58 xy_prime = zeros(size(temp));
59 for kk=1:length(c)
60     indices=c(kk,:);
61     xpos = mod(kk-1,2)+1;
62     ypos = mod(floor((kk-1)/2),2)+1;
63     if ([xpos ypos] == [1 1])
64         twosquare = zeros(2);
65     end
66     twosquare(xpos, ypos) = temp(indices(1),indices(2));
67     if (mod(kk,4) == 0)
68         j = kk/4;
69         xy_prime(:,j) = twosquare;
70     end
71 end
72 xy_prime=xy_prime(:,1:j);
73 %
74 % %% %%% %%%% %%%%% %%%%%% %%%%%%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%%%%%
75 end

```

---

To extract the streaks first the edges in the image are detected. This is done using the *MATLAB* *edge* function. *MATLAB* describes the *edge* function as taking a binary image as its input, and returning a binary image of the same size, with 1's where the function finds edges and 0's elsewhere. The *canny* method, finds edges by looking for local maxima of

the gradient, was chosen because it out performed other methods during the verification process.

Lines 18-24 determine and display the Hough Transform. Line 19 uses the *MATLAB* *hough* function, which implements the Standard Hough Transform. *MATLAB* describes the function as being designed to detect lines using the parametric representation of a line:

$$\rho = x \cos(\theta) + y \sin(\theta) \quad (\text{A.1})$$

Where the variable  $\rho$  is the distance from the origin to the line along a vector perpendicular to the line, and  $\theta$  is the angle between the x-axis and this vector. Lines 20-24 then graph the Hough Transform matrix, as shown in Figure A.8.

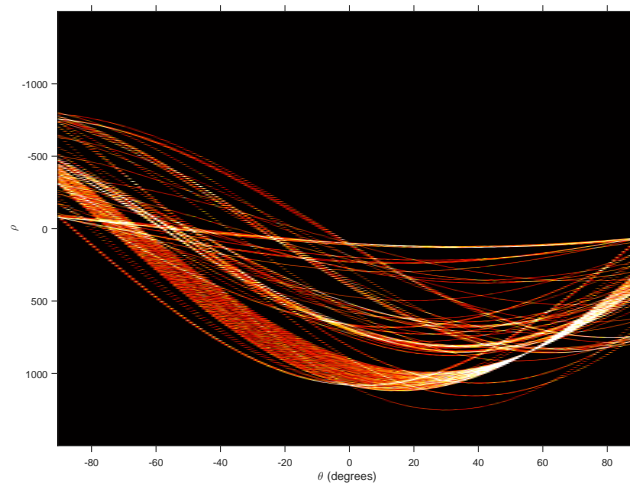


Figure A.8: The Hough Transform of the binary streak image

Once the Hough Transform matrix has been determined line 28 uses the *matlab* *houghpeaks* function to find the peaks in the Hough Transform matrix, defining both the number of peaks and the threshold those peaks must cross to be considered. Lines 29-32 then superimposes a plot of these peak locations onto the image of the transform.



Line 34 then uses the *matlab houghlines* function to find the lines in the image, defining both the size of gaps which may be ignored and the minimum line length. Both of these numbers are large here because the function was last used to process unbound streak data as described in Chapter 4. Once the lines in the image have been identified lines 36-53 of Listing A.3 superimpose those lines back onto the original fits file as shown in Figure A.9. The longest line in the image is identified on line 49 and returned by the function. Finally, lines 55-72 are used to extract pixel locations of other streaks in the image from the array they are located in and place them into the matrix `xy_prime` to also be returned by the function.

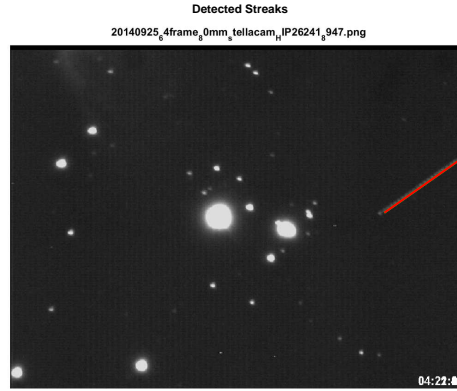


Figure A.9: The identified RSO streak

## A.2 Convert Pixels to Spherical $\alpha$ and $\delta$

Once pixel location for the streaks has been determined those pixels need to be transformed into usable position vectors. The first step is to average the end point pixel location for each streak to determine the center location of the pixel, see Section 4.7.1 for why this is important. Then the image and the pixel location (`xy_prime`), can be

given to *astrometry.net*. As discussed in Chapter 4, *astrometry.net* is used to perform an astronomical calibration on the original image, see Figure A.2. The tool identifies the background star field and returns the astronomical WCS meta-data including the principal point, pixel scale, and the rotation of the image[58]. Each of these pieces of information is required to find the correct spherical coordinates of the RSO, and create the WCS information returned in the header of the new fits file. The WCS provides the pixel space to a real-world position transformation to ECI J2000 coordinates, allowing the pixel location of an RSO in the image to directly provide the angles data required for orbital information to be derived. For a detailed verification and validation of the *astrometry.net* tool see Reference 58.

### **A.3 Right Ascension, Declination, and Time**

Now, the correct right ascension and declination have been returned. The time stamp can be pulled directly from the fits header and the required information for a B3 report used by AFSPC has been determined. Angles determined using this method are used to provide an orbital update using additional methods, see Chapter 4 and Appendix C.

## Appendix B: Verification and Validation of Angle Extraction Techniques

### B.1 Transforming Camera Model Position Vectors into the SEZ Reference Frame

To perform the verification and validation of the models compared in Chapter 3 a star field was used to ensure proper application of each method. To ensure an accurate comparison the output of each method was transformed into a sensor reference frame. For the affine transform method the reference star locations were provided in the traditional SEZ coordinate system, returning other star positions in the same coordinate system. Both camera models had to be transformed from the normalized  $[XYZ]^T$  vector to the SEZ sensor reference frame requires both a rotation to line up the coordinate systems and a translation to co-locate the system origins. Note that the camera model reference frame is not the same for the pinhole camera model as it is for the intrinsic calibration model. Therefore each system will require its own rotation matrix as given below in Sections B.1.1 and B.1.2.

#### B.1.1 Pinhole Camera Model Position Vector to the SEZ Reference Frame

To transform the  $[X \ Y \ Z]^T$  position vector into the SEZ frame requires rotating about the Y axis by  $(90^\circ - el_0)$ , and the intermediate Z axis by the azimuth  $(\beta_0)$  [13]. Figure B.1 provides an example of how these systems might be misaligned. To align these systems first rotate around the Y axis using Equation B.1 [8].

$$ROT1(90^\circ - el_0) = \begin{bmatrix} \cos(90^\circ - el_0) & 0 & -\sin(90^\circ - el_0) \\ 0 & 1 & 0 \\ \sin(90^\circ - el_0) & 0 & \cos(90^\circ - el_0) \end{bmatrix} \quad (B.1)$$

This rotation brings the zenith and z-axis into alignment as shown in Figure B.3.

Rotating to align the X axis with the east direction and the Y axis with the south direction can be accomplished with a rotation around Zenith equal to  $\beta_0$  as demonstrated in

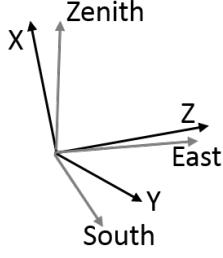


Figure B.1: Offset between systems

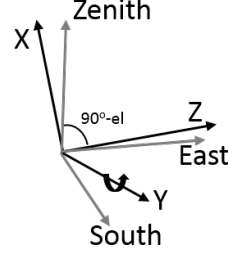


Figure B.2: Aligning the Zenith/Z axis

Figure B.3. Mathematically this is done using the rotation matrix defined by Equation B.2 [8].

$$ROT3(\beta_0) = \begin{bmatrix} \cos(\beta_0) & \sin(\beta_0) & 0 \\ -\sin(\beta_0) & \cos(\beta_0) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (B.2)$$

With the final rotation complete the axis of both systems are aligned as shown in Figure B.4. Therefore this 2-3 rotation provides a position vector  $\rho$  in the SEZ coordinate system when applied to the  $[XYZ]^T$  vector as shown in Equation B.3.

$$\begin{bmatrix} S \\ E \\ Z \end{bmatrix} = ROT3(\beta_0) ROT2(90^\circ - el_0) \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (B.3)$$

### ***B.1.2 Intrinsic Calibration Model Position Vector to the SEZ reference frame***

Figure B.5 provides an example of how the intrinsic calibration model and the SEZ coordinate systems might be misaligned. To assure alignment of the systems a 1-3 rotation is required. First rotate around the x-axis by  $90^\circ - el$  as demonstrated by Figure B.6. Mathematically this is done using the rotation matrix defined by Equation B.4 [8]. This rotation brings the zenith and z-axis into alignment as shown in Figure B.6.

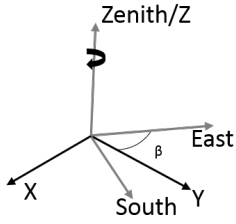


Figure B.3: Aligning East/X, South/Y

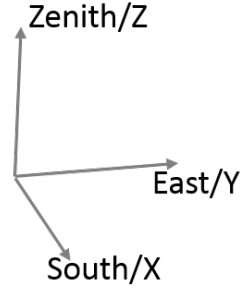


Figure B.4: The aligned system

$$ROT1(90^\circ - el_0) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(90^\circ - el_0) & \sin(90^\circ - el_0) \\ 0 & -\sin(90^\circ - el_0) & \cos(90^\circ - el_0) \end{bmatrix} \quad (B.4)$$

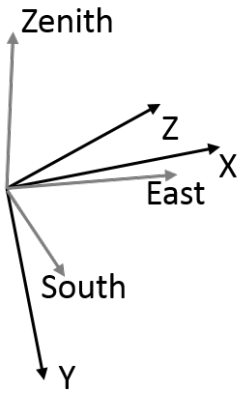


Figure B.5: Offset between two systems

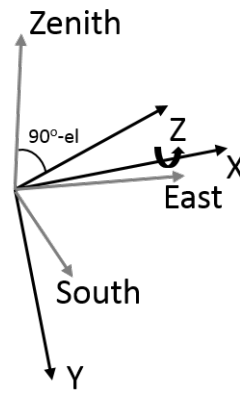


Figure B.6: Aligning the Zenith/Z axis

Rotating to align the  $X$  axis with the east direction and the  $Y$  axis with the south direction can be accomplished with a rotation around Zenith equal to  $270^\circ + \beta$  as demonstrated in Figure B.7. Mathematically this is done using the rotation matrix defined

by Equation B.2 [8]. The azimuth used here is a reference to the bore sight angle and not a tracked satellite.

With the final rotation complete the axis of both systems are aligned as shown in Figure B.8. Therefore this 1-3 rotation provides a position vector  $\rho$  in the SEZ coordinate system when applied to the  $[XYZ]^T$  vector as shown in Equation B.5.

$$\begin{bmatrix} S \\ E \\ Z \end{bmatrix} = ROT3(\beta_0) ROT1(90^\circ - el_0) \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (B.5)$$

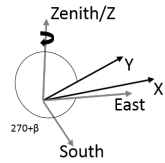


Figure B.7: Rotation about 3<sup>rd</sup> Axis

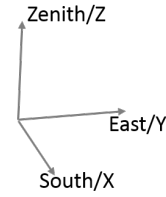


Figure B.8: The aligned system

### B.1.3 Spherical Coordinate Transformation

Once the position vector is defined in the SEZ coordinate system a simple spherical transformation is required to determine the look angle (azimuth and elevation). First the azimuth of the tracked satellite can be calculated using Equation B.6 [93].

$$\beta = \arctan\left(\frac{\rho_y}{\rho_x}\right) \quad (B.6)$$

Note that a two argument arc-tangent is required to assign the result to correct quadrant. Otherwise the quadrant will need to be assigned manually using the following definitions:  $\frac{+}{+} \rightarrow$  first;  $\frac{+}{-} \rightarrow$  second;  $\frac{-}{-} \rightarrow$  third;  $\frac{-}{+} \rightarrow$  fourth; to calculate the elevation use Equation B.7 [93].

$$el = \arcsin\left(\frac{-\rho_z}{|\rho|}\right) \quad (\text{B.7})$$

#### ***B.1.4 Translating the Position Vector***

The origin of the SEZ coordinate frame is actually located inside the sensor or CCD itself, coinciding the origin with the  $x_s, x_y$  system. Therefore to make the camera model position vector coincide with the SEZ system the origin must shift to align the principal point with its position in real space. This translation is accomplished by determining the difference between the reported angle of the principle point and several other stars, and the knowledge of the objects true position and then adding the average difference to the spherical coordinates returned by Equations B.6 and B.7 as shown in Equation B.8.

$$\begin{bmatrix} \beta_{translated} \\ el_{translated} \end{bmatrix} = \begin{bmatrix} \beta \\ el \end{bmatrix} + \begin{bmatrix} T_\beta \\ T_{el} \end{bmatrix} \quad (\text{B.8})$$

## **B.2 Verification and Validation of Astrometry Tools**

To ensure that the models work as intended a brief verification and validation was performed. The models are validated as a part of the verification process, each model should return pixel specific look angles given the appropriate inputs. To verify that the models work two verification steps are used. First, a white box verification is accomplished. Second, the correct azimuth and elevation for 20 stars with known angles were processed and the output compared to truth.

### ***B.2.1 White Box Verification and MATLAB Implementation***

Implementing the mathematical theory outlined in Sections 2.4.1, 2.4.2, and 2.4.3 will be performed using MATLAB scripts and function calls. To ensure these scripts and functions perform as expected, a white box verification of each script and function will be performed. This verification will identify each equation in the sections listed above to a line of code in the listings given below.

The scripts for each method provided in the sections listed above will be given in Section B.2.1.1, B.2.1.2, and B.2.1.3. Each function called by these scripts is provided in Section B.2.2.

### ***B.2.1.1 MATLAB Affine Transform Astrometry White Box Verification***

The method for performing an astrometry using an affine transform is provided in Section 2.4.1. This method is implemented in MATLAB using the script provided in Listing B.1, given below.

Listing B.1: Affine transform model source code

---

```

1 % This script is designed to allow for affine transform processing of an
2 % image:
3
4 %Note: you will need to load a table of the extracted pixel locations for
5 %objects of interest.
6 %Note: the program will return a look angle in the same coordinate
7 %system it is provided reference angles.
8
9 %% Given:
10 FOV=[1.402*pi/180 1.057*pi/180]; %degrees to radians
11 boresight=[214.1906*pi/180 38.241944*pi/180];
12 %beta stars=213 34'51" 213 44'02" 214 46'37"
13 beta_stars=[213.580833*pi/180; 213.733889*pi/180; 214.776944*pi/180];
14 %el stars=38 25'32" 37 38'40" 38 47'37"
15 el_stars=[38.425556*pi/180; 37.644444*pi/180; 38.793611*pi/180];
16 ref_pixels=[29.612903225806452 2.364193548387097e+02;...
17             93.870129870129870 5.983246753246754e+02;5.461363636363636e+02 66];
18
19 %% Perform an affine transform to produce the observed angles:
20 az_el = affine_transform( boresight(1,1),boresight(1,2), beta_stars,...
21                           el_stars, ref_pixels, streak_pixels);

```

---

In the script provided line 10 is used to provide the systems FOV. Line 11 provides the bore sight azimuth and elevation. Lines 13 and 15 provide the azimuth and elevation of the reference stars in the image. Lines 16-17 provide the pixel values for those reference stars. The “affine\_transform” function, called on line 20, is provided in Section B.2.2.1.

### ***B.2.1.2 MATLAB Pinhole Camera Model White Box Verification***

With the understanding of how to use the pinhole camera model, provided in Section 2.4.2, the process is implemented MATLAB using the script provided in Listing B.2.



## Listing B.2: Pinhole camera model source code

---

```

1  % This script performs several calculations and function calls to run a
2  % table of pixel values through a pinhole camera model and return the
3  % observed azimuth and elevation of the pixel locations in the image.
4
5  % You will need to import a table of pixel locations before running.
6
7  %% Given:
8  % Read in an image:
9  originalImage=imread('C:\Images\matlab_images\baseline_centroids.png');
10 %look angles of Ox Vir converted from degrees to radians
11 boresight=[214.1906*pi/180 38.241944*pi/180];
12 % other system properties
13 FOV=[ 1.402*pi/180 1.057*pi/180];
14
15 %% Now we need to process the image and extract som values:
16 % we require the size of the image in pixels:
17 % V H -
18 [M, N, d]=size(originalImage);
19 % Find the defined variables for the function calls below
20 %Determine the center points:
21 O_x=(M+1)/2;
22 O_y=(N+1)/2;
23 % find virtual focal length f'
24 f_prime=N/(2*atan(FOV(1,1)/2));
25
26 %% next create the "XYZ" reference frame position vector:
27 d=size(streak_pixels);
28 for k=1:d(1,1)
29     XYZ(:,k) = ph_XYZ( O_x,O_y,streak_pixels(k,1),streak_pixels(k,2),...
30         f_prime );
31 end
32
33 %% Then transform the XYZ coordinate system to the SEZ coordinate system
34 %streak 1
35 SEZ = ph_XYZ_SEZ( boresight(1,1),pi/2-boresight(1,2),XYZ );
36
37 %% Convert the above LOS vector into azimuth and elevation spherical
38 % coordinates.
39 for k=1:d(1,1)
40     [ AZ(k,:),EL(k,:) ] = az_el_third( SEZ(:,k) );
41 end
42
43 %% Detect the translation present in the new image:
44 translation(1,1)=(sum(true_location(:,1)-AZ(:,1)))/d(1,1);
45 translation(1,2)=(sum(true_location(:,2)-EL(:,1)))/d(1,1);
46 % Translate the position vectors:
47 for k=1:d(1,1)
48     [ AZEL(k,:) ] = [AZ(k,:),EL(k,:)]+translation;
49 end

```

---

Line nine is used to read in the given image. The given values are input into lines 11-13. Line 11 provides the bore sight azimuth and elevation, and line 13 provides the FOV of the system.

Lines 16-24 are used to process the given image to get required information. Line 18 determines the number of pixels in the x and y direction. To continue MATLAB needs to take the given information and determine the center point of the image ( $O_x$ ,  $O_y$ ) and the virtual focal length ( $f'$ ). Lines 21-22 are used implement Equation 2.10 to provide the center point of the image. Line 24 implements Equation 2.11 to determine the virtual focal length.

Lines 26-41 are used to call several functions discussed in Section B.2.2. These functions loop through each provided pixel location and transform them into azimuth and elevation angles as discussed in Section B.1.3.

The section beginning on line 26 uses a for loop to create a position vector for each pixel provided in the Pinhole camera model reference frame. Lines 28-31 loop through the function “ph\_XYZ” , provided in Section B.2.2.2, to define the position vector.

The next section, beginning on line 33, performs the transformation to the SEZ coordinate system by calling the function “ph\_XYZ\_SEZ” on Line 35. The function “ph\_XYZ\_SEZ” is provided in Section B.2.2.3.

The final function call, beginning on line 37, performs the transformation to spherical coordinates. Lines 39-41 loop through the function “az\_el\_third” , provided in Section B.2.3, to perform the transformation for each position vector provided.

Although the rotation is completed by the “ph\_XYZ\_SEZ” on line 40, see Section B.2.5, and the position vector translated into spherical coordinates by line 40, see Section B.2.3, the position is still located at the origin of the pinhole camera model. To translate to the correct real world position the position vector requires the translation given by Equation B.8. Lines 44 and 45 determine the required translation by averaging the residuals for both azimuth and elevation of the known stars in the image. Lines 47-49 loop through the determined translation. Each angle is translated as defined on line 48 to shift to the correct position.

### B.2.1.3 Intrinsic Parameter Calibration MATLAB Model

The intrinsic parameter calibration model, provided in Section 2.4.3, is implemented in MATLAB in a fashion similar to the pinhole camera model. This is expected as they are both perspective projection models [56]. The full script is available below in Listing B.3.

Listing B.3: Camera calibrated with intrinsic parameter Model Source Code

---

```
1 % This script performs several calculations and function calls to run a
2 % table of pixel values through a camera with known intrinsic parameters
3 % model and return the observed azimuth and elevation of the pixel
4 % locations in the image.
5
6 % You will need to import a table of pixel locations before running.
7
8 %% Given:
9 % Read in an image:
10 originalImage=imread('C:\Images\matlab_images\baseline_centroids.png');
11 %look angles of 0x Vir converted from degrees to radians
12 boresight=[214.1906*pi/180 38.241944*pi/180];
13 % other system properties
14 FOV=[ 1.402*pi/180 1.057*pi/180]; % Field of view
15 %Define the scaling factors:
16 s_x=1.13;
17 s_y=0.87;
18
19 %% Now we need to process the image and extract some values:
20 % we require the size of the image in pixels:
21 % V H -
22 [M, N, d]=size(originalImage);
23 % Find the defined variables for the function calls below
24 %Determine the center points:
25 O_x=(M+1)/2;
26 O_y=(N+1)/2;
27 % find virtual focal length f'
28 f_prime(1,1)=N/(2*atan(FOV(1,1)/2));
29 f_prime(1,2)=M/(2*atan(FOV(1,2)/2));
30
31 %% next create the "XYZ" reference frame position vector:
32 d=size(streak_pixels);
33 for k=1:d(1,1)
34     XYZ(:,k) = simple_IPM(streak_pixels(k,1), streak_pixels(k,2), O_x,...
35         O_y, s_x, s_y, f_prime );
36 end
37
38 %% Then transform the XYZ coordinate system to the SEZ coordinate system
39 %streak 1
40 SEZ = XYZ_SEZ( boresight(1,1),pi/2-boresight(1,2),XYZ );
41
42 %% Convert the above LOS vector into azimuth and elevation spherical
43 % coordinates.
44 for k=1:d(1,1)
45     [ AZ(k,:),EL(k,:)] = az_el_third( SEZ(:,k) );
46 end
47
48 %% Detect the translation present in the new image:
49 translation(1,1)=(sum(true_location(:,1)-AZ(:,1)))/d(1,1);
```

```

50 translation(1,2)=(sum(true_location(:,2)-EL(:,1)))/d(1,1);
51
52 % Translate the position vectors:
53 for k=1:d(1,1)
54     [ AZEL(k,:) ] = [AZ(k,:),EL(k,:)]+translation;
55 end

```

---

Line 10 is used to read in the image that we are given. The given information is provided on lines 12-17. Line 12 provides the bore-sight azimuth and elevation for the center of the image. Line 14 provides the FOV covered by the image. Finally, the scale factor is defined on lines 16-17.

Lines 20-29 are used to process the given image to get required information. Line 22 determines the number of pixels in the x and y direction. To continue MATLAB needs to take the given information and determine the center point of the image ( $O_x$ ,  $O_y$ ) and the virtual focal length ( $f'$ ) in both the vertical and horizontal direction. Lines 25-26 are used implement Equation 2.10 to provide the center point of the image. Lines 28-29 implements Equation 2.11 to determine the virtual focal length.

Lines 31-46 are used to call several functions discussed in Section B.2.2. These functions loop through each provided pixel location and transform them into azimuth and elevation angles as discussed.

The section beginning on line 31 uses a for loop to create a position vector for each pixel provided in the camera model reference frame. Lines 33-36 loop through the function “simple\_IPM”, provided in Section B.2.4, to define the position vector.

The next section, beginning on line 38, performs the transformation to the SEZ coordinate system by calling the function “XYZ\_SEZ” on Line 40. The function “XYZ\_SEZ” is provided in Section B.2.3.

The next section, beginning on line 42, performs the transformation to spherical coordinates. Lines 44-46 loops through the function “az\_el\_third” , provided in Section B.2.3, to perform the transformation.

Although the rotation is complete and the position vector translated into spherical coordinates, the position is still located at the origin of the calibrated camera model system. To translate to the correct real world position the position vector requires the translation given by Equation B.8. Lines 49 and 50 determine the required translation by averaging the residuals for both azimuth and elevation of the known stars in the image. Lines 53-55 loop through the determined translation. Each angle is translated as defined on line 54 to shift to the correct position using Equation B.8.

### ***B.2.2 White Box Verification of Functions***

The scripts given above perform several function calls to solve parts of the problem. Those function calls are discussed below. To verify each function call it is compared to the theory behind its creation covered in Sections 2.4.1, 2.4.2, and 2.4.3.

#### ***B.2.2.1 Performing Affine Transform Astrometry***

Listing B.1 provides all of the required information to perform an affine transform on an image. Line 20 of Listing B.1 passes this information to Listing B.4. Listing B.4 implements the process provided in Section 2.4.1.

Listing B.4: Performing an affine transform

---

```

1 function [ alpha_dec ] = affine_transform( alpha_0,dec_0, alpha, dec, xy, streak )
2 %% Affine Transform Astrometry approach for finding streak location in the
3 % sky
4
5 %% Calculate sky coordinates for reference stars
6 d=size(xy);
7 for k=1:d(1,1)
8     X(k,:)=(cos(dec(k,1))*sin(alpha(k,1)-alpha_0))...
9             /(cos(dec_0)*cos(dec(k,1))*cos(alpha(k,1)-alpha_0)...
10             +sin(dec_0)*sin(dec(k,1)));
11     Y(k,:)=(-1)*((sin(dec_0)*cos(dec(k,1))*cos(alpha(k,1)-alpha_0)...
12             -cos(dec_0)*sin(dec(k,1)))...
13             /(cos(dec_0)*cos(dec(k,1))*cos(alpha(k,1)-alpha_0)...
14             +sin(dec_0)*sin(dec(k,1))));
15 end
16
17 %% Solve for the plate constants
18 A=zeros(d(1,1),3);
19 for k=1:d(1,1)
20     A(k,1)=xy(k,1);
21     A(k,2)=xy(k,2);
22     A(k,3)=1;
23 end

```

```

24 % in the X direction
25 plate_x=rref([A X]);
26 % in the Y direction
27 plate_y=rref([A Y]);
28
29 %% calculate sky coordinates of each streak
30 d=size(streak);
31 X_streak=zeros(d(1,1),1);
32 Y_streak=zeros(d(1,1),1);
33 for k=1:d(1,1)
34     X_streak(k,:)=plate_x(1,4)*streak(k,1)+plate_x(2,4)*streak(k,2)+...
35         plate_x(3,4);
36     Y_streak(k,:)=plate_y(1,4)*streak(k,1)+plate_y(2,4)*streak(k,2)+...
37         plate_y(3,4);
38 end
39
40 %Find declination:
41 dec_streak=asind((sin(dec_0)+Y_streak*cos(dec_0))...
42     ./sqrt(1+X_streak.^2+Y_streak.^2));
43 %Find RA:
44 alpha_streak=alpha_0*180/pi+atand(X_streak...
45     ./((cos(dec_0)-Y_streak*sin(dec_0))));
46
47 alpha_dec(:,1)=alpha_streak(:,1);
48 alpha_dec(:,2)=dec_streak(:,1);
49
50 end

```

---

The function “affine\_transform” takes the inputs *alpha\_0*, *dec\_0*, *alpha*, *dec*, *xy* and *streak* and returns the look angle for each pixel present in the vector “streak”. Lines 5-15 loop through Equations 2.2 and 2.3 to determine the sky coordinates for each of the reference stars. Lines 17-27 are used to determine the plate constants by setting up a matrix and finding its reduced row echelon form, to solve Equation 2.6, providing the plate constants. Once the plate constants are known Lines 33-38 loop through Equations 2.4 and 2.5, to solve for the sky coordinates of each pixel in the matrix “streak”. Finally Lines 40-48 calculate the correct look angles for each pixel. Specifically line 41 solve Equation 2.7 and line 44 solves Equation 2.8. Lines 47 and 48 simply place the returned values into a single matrix. Finally, Line 50 terminates the function call and returns the matrix of desired look angles.

### ***B.2.2.2 Finding the the Pinhole Camera Model Position Vector***

Lines 27-31 of Listing B.2 loop each table entry through the function: “simple\_IPM” provided in Listing B.5. Listing B.5 implements Equation 2.12, provided in Section 2.4.2.

### Listing B.5: Determining the pinhole camera model vector

---

```

1 function [ XYZ ] = ph_XYZ( O_x,O_y,y_prime,x_prime,f_prime )
2 %Takes provided inputs and returns the pinhole normalized position vector
3 % This function takes inputs x', y', O_x, O_y, and f and
4 % returns a pinhole model normalized position vector using
5 % x', y' are the pixel locations of the object we desire to locate in
6 % real space
7 % O_x, O_y are the center pixel coordinates of the image.
8 % f' is the focal length of the camera system
9 X=O_x-x_prime;
10 Y=y_prime-O_y;
11 Z=f_prime;
12 XYZ=[X; Y; Z];
13 end

```

---

The function “ph\_XYZ” takes the inputs  $x'$ ,  $y'$ ,  $O_x$ ,  $O_y$ , and  $f'$  and returns the pinhole normalized position vector. Lines 9-11 solve for the first, second, and third entry of Equation 2.12 respectively. Line 13 terminates the function call and returns the  $[XYZ]^T$  vector.

#### ***B.2.2.3 Rotating the Pinhole Camera Model Vector***

The section beginning on line 33 of Listing B.2, implements the rotation discussed in Section B.1.1. Line 35 calls the function “ph\_XYZ\_SEZ”, provided in Listing B.6 to rotate the position vectors to align it with the SEZ vector.

### Listing B.6: Rotating the camera model vector to SEZ

---

```

1 function [ SEZ ] = ph_XYZ_SEZ( theta,phi,XYZ )
2 %Rotates a normalized position vector to align it with SEZ
3 % This function takes inputs phi, theta, and the vector XYZ and returns
4 % the SEZ vector after performing a 2-3 rotation on it.
5 % phi is the angle of the first rotation in radians
6 % theta is the angle of the second rotation in radians
7 ROT2 = [cos(phi) 0 -sin(phi);...
8         0 1 0; sin(phi) 0 cos(phi)];
9 ROT3 = [cos(theta) sin(theta) 0;...
10        -sin(theta) cos(theta) 0; 0 0 1];
11 SEZ = ROT3* ROT2* XYZ;
12 end

```

---

The function “ph\_XYZ\_SEZ” takes inputs  $\theta$ ,  $\phi$ , and the vector  $[XYZ]^T$  and returns the SEZ vector after performing a 2-3 rotation on it. Lines 7-8 defines the rotation

given by Equation B.1, and lines 9-10 define the rotation given by Equation B.2. Finally line 11 performs the 2-3 rotation, given by Equation B.3, on the provided vector and line 12 terminates the function call and returns the  $[SEZ]^T$  vector.

### ***B.2.3 Transforming into Spherical Coordinates***

Both Listing B.2 and Listing B.3, perform the transformation to spherical coordinates. This is accomplished through the function “az\_el\_third” to perform the transformation.

The function “az\_el\_third”, provided in Listing B.7, takes the provided SEZ vector and returns the spherical coordinates  $(\beta, el)$ . Line 9 implements Equation B.6. Line 10 implements Equation B.7. Finally line 11 terminates the function call and returns the  $\beta$  and  $el$  angles.

Listing B.7: Transforming SEZ to look angle

---

```

1 function [ AZ, EL ] = az_el_third( rho )
2 %Takes provided inputs and returns the spherical coordinates (AZ, EL)
3 % This function takes input vector rho and returns the azimuth and
4 % elevation associated with the provided vector.
5 % rho is a position vector in the SEZ coordinate system.
6 % Az is the azimuth of the angle returned in degrees
7 % el is the elevation of the angle returned in degrees
8
9 AZ=atand(rho(2,1)/rho(1,1))+180;    %(-(-))/+ is third quad +180
10 EL=asind(-rho(3,1)/norm(rho));
11 end

```

---

### ***B.2.4 Finding the Calibrated Camera Model Position Vector***

Lines 33-36 of Listing B.3, loop through a function, “simple\_IPM” provided in Listing B.8. Listing B.8 implements Equation 2.14, provided in Section 2.4.3.

Listing B.8: Determining the calibrated camera vector

---

```

1 function [ XYZ ] = simple_IPM( x_prime, y_prime, O_x, O_y, s_x, s_y, f )
2 %Takes provided inputs and returns the simplified answer position vector
3 % This function takes inputs x', y', O_x, O_y, s_x, s_y, and f and
4 % returns a skew free normalized position vector using a camera with
5 % intrinsic parameters model.
6 % x', y' are the pixel locations of the object we desire to locate in
7 % real space
8 % O_x, Y_x are the center pixel coordinates of the image.
9 % s_x, s_y are the scale factors for the image

```

---



```

10 % f is the focal length of the camera system
11
12 X=(x_prime-0_x)/(s_x*f);
13 Y=(y_prime-0_y)/(s_y*f);
14 Z=1;
15
16 XYZ=[X; Y; Z];
17
18 end

```

---

The function “simple\_IPM” takes the inputs inputs  $x'$ ,  $y'$ ,  $O_x$ ,  $O_y$ ,  $s_x$ ,  $s_y$ , and  $f$  and returns a skew free normalized position vector using a camera with intrinsic parameters model. Line one initializes the function. Lines 2-10 provide a description of the function and how it works. Lines 12, 13, and 14 solve for the first, second, and third entry of Equation 2.14 respectively. Line 18 terminates the function call and returns the  $[XYZ]^T$  vector.

### ***B.2.5 Rotating the Calibrated Camera Model Vector***

Listing B.3 calls the function “XYZ\_SEZ” to implement the rotation discussed in Section B.1.2 to align the position vector with the SEZ vector. The function “XYZ\_SEZ”, provided in Listing B.9, takes inputs phi, theta, and the vector XYZ and returns the SEZ vector after performing a 1-3 rotation on it. Lines 9-10 define the rotation given by Equation B.4, and lines 11-12 define the rotation given by Equation B.2. Finally line 14 performs the 1-3 rotation, given by Equation B.5, on the provided vector and line 16 terminates the function call and returns the  $[SEZ]^T$  vector.

Listing B.9: Rotating the calibrated camera vector to SEZ

---

```

1 function [ SEZ ] = XYZ_SEZ( phi,theta,XYZ )
2 %Rotates a normalized position vector to align it with SEZ
3 % This function takes inputs phi, theta, and the vector XYZ and returns
4 % the SEZ vector after performing a 1-3 rotation on it.
5 % phi is the angle of the first rotation in radians
6 % theta is the angle of the second rotation in radians
7
8 %Convert Normalized Coordinates to SEZ coordinates:
9 ROT1 = [1 0 0; 0 cos(phi) sin(phi);...
10         0 -sin(phi) cos(phi)];
11 ROT3 = [cos(theta) sin(theta) 0;...
12         -sin(theta) cos(theta) 0; 0 0 1];
13

```

```

14 SEZ = ROT3* ROT1* [XYZ];
15
16 end

```

---

### ***B.2.5.1 20 Star Verification***

The Second Verification process compares the expected look angles of 20 stars with known look angles. This was accomplished by generating a noise free simulated image of a small FOV of the night sky. The image was processed to extract the pixel locations of the known stars, and those pixel locations were used as inputs to each astrometry tool. The results, provided graphically can be seen in Figures B.9, B.10, and B.11 are directly compared to the true look angle of each star. Each model is discussed in more detail below.

The results of the affine transform verification are shown in Figure B.9, again directly overlying the true positions for visual comparison. As can be seen in Figure B.9 the overlay of the true position on the returned values is almost an exact match. As the affine transform method attempts to account for all distortion this outcome is as expected. Figure B.9 further verifies that the MATLAB script provided in Listing B.1, is functioning as expected.

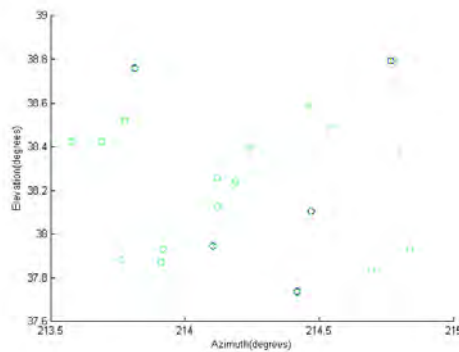


Figure B.9: Affine transform astrometry overlying true positions

The results of the pinhole camera verification are shown in Figure B.10, again overlying the true positions, for visual comparison. As shown in Figure B.10, the pinhole

camera results are best near the center of the image and deviate, mostly in elevation, as we deviate from center. This focal length distortion is not unexpected as the pinhole model only accounts for the focal length in the horizontal direction, and unlike a model that includes intrinsic parameters, does not account for scale in the image. Scale is present even in the noise free image as the CCD pixels are rectangular and not perfectly square. As expected the further from the principle point we get the more the scale affects the image. As the distortions in the returned positions can be accounted for in the model, Figure B.10 further verifies that the MATLAB script provided in Listing B.2 is functioning correctly.

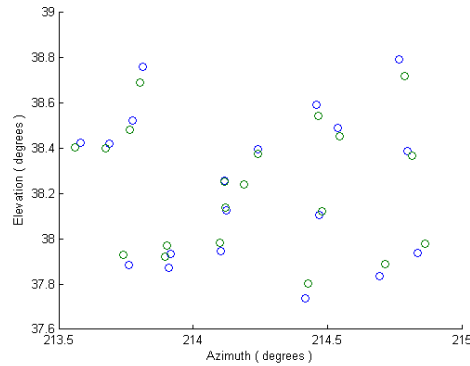


Figure B.10: Pinhole camera model overlying true positions

Finally, the results of the intrinsic parameter calibrated camera model verification are shown in Figure B.11, again overlying the true positions, for visual comparison. For the intrinsic camera calibrated image the overlay of the true position on the returned values is almost an exact match. As the calibrated image accounts for all of the camera distortion you would expect this result to be true for a noise free image. Figure B.11 further verifies that the MATLAB script provided in Listing B.3, is functioning as expected.

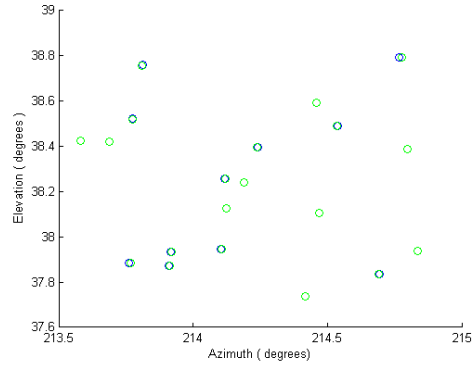


Figure B.11: Intrinsic parameters calibrated model overlying true positions

### ***B.2.6 Validation of the Angle Extraction Tools***

To validate each of the astrometry method scripted in Section B.2.1 requires confirmation that when the required inputs are given, look angles for the desired pixel locations are returned. This was demonstrated in Section B.2.5.1 as each method was verified. Therefore, further validation of the models is not required.

## Appendix C: Orbital Determination Unscented Kalman Filter

The focus of this dissertation was on the extraction of angles data from images provided by external sensors. The method developed isolated streaks and extracted angles data as described in Appendix A. The intention was to be able to pass the angles data to an existing orbital algorithm including: Laplace, Gauss, and the Double-R method using standard orbital determination algorithms [8].

When the determination of orbital data was required for Phase II Gary Goff's UKF algorithm was used. This appendix describes how that algorithm was implemented for the work presented here. The majority of this Appendix was written by Gary Goff as his contribution to the Paper: Updating Position Data from Full Serendipitous Satellite Streaks.

### C.1 Orbital Determination UKF

When performing an orbit estimation, a key early decision is to determine an element set to represent both the orbital state and the covariance. An orbital solution is composed of  $n = 6$  components, where  $n$  is the size of the state vector. A traditional choice is Earth-centered Cartesian coordinates of position and velocity, as the covariance is easy to visualize. Recent works by the team at Numerica, Inc. have shown in several papers that equinoctial elements better preserve the underlying error distribution [73, 94, 95]. Based on these works and initial testing by the authors, an equinoctial element UKF was implemented to process telescope observations while estimating the orbital state and covariance. The state is define as a set of equinoctial elements [8].

$$\mathbf{x} = \begin{bmatrix} a & h & k & p_e & q_e & \ell \end{bmatrix}^T \quad (\text{C.1})$$

Unfortunately, the elements are not easily visualized like classical orbital elements, but equinoctial elements are often used to avoid singularities present in classical orbital elements [96]. The estimated covariance matrix for the corresponding state is captured in

the matrix  $\mathbf{P}$ . The UKF routine requires an initial state and covariance which is provided from a previous estimate or determined via initial orbit determination routines for newly discovered RSOs.

As mentioned in the background section, the UKF implements the UT which uses weighted sampled sigma points (specifically  $2n+1$ ) to estimate the nonlinear transformation [97]. The symmetric sigma point weighting approach is commonly used and the optimal  $\kappa$  value for Gaussian distributions is defined such that  $n + \kappa = 3$  [98]. The mean and covariance weights of sigma point  $j$  are defined as:

$$w_m^0 = w_c^0 = \frac{\kappa}{n + \kappa} \quad (\text{C.2})$$

$$w_m^{(j)} = w_c^{(j)} = \frac{1}{2(n + \kappa)} \quad \text{for } j = 1, \dots, 2n \quad (\text{C.3})$$

where  $\kappa = -3$  is recommended for orbital elements.

The UT requires taking the square root of the covariance matrix. For a symmetric, positive definite matrix (true for orbital covariance matrices), this operation is easily performed via a Cholesky decomposition [99]. Implementing the UT into the UKF for orbit estimation requires two main steps: propagate and update. The detailed steps of the orbital estimation UKF are provided in Algorithm 1 [97].

The UKF in Algorithm 1 requires propagating each state sigma point using some form of orbital state propagation (General or Special Perturbation). For this research, Microcosm's HPOP propagator is implemented due to its speed and accuracy [100]. Once all sigma points are propagated, the propagated covariance,  $\bar{\mathbf{P}}$ , is estimated within the UKF. The process noise covariance,  $\mathbf{Q}$ , captures the distribution of propagation errors. Due to relatively short propagation times and orbit fitting test results,  $\mathbf{Q}$  was set to zero as the propagation step did not induce significant errors. The actual observation of the state is performed at the telescope when it reports a right ascension,  $\alpha$ , and declination,  $\delta$ , angle. The actual observation vector is defined such that  $\mathbf{y} = \begin{bmatrix} \alpha & \delta \end{bmatrix}^T$ . The telescope's errors are assumed to be an uncorrelated Gaussian with a covariance captured in  $\mathbf{R}$  such

---

**Algorithm 1:** Unscented Kalman Filter
 

---

**Given:**  $\kappa$ ,  $w_m^j$  and  $w_c^j$  for  $j = 1, \dots, 2n$

- 1 Define or update previous reference:  $\hat{\mathbf{P}}_0, \hat{\mathbf{x}}_0$
- 2 Read in the next observation:  $t_i, \mathbf{y}_i, \mathbf{R}_i$
- 3 Perform decomposition  $\mathbf{P}_{i-1} = \mathbf{A}^T \mathbf{A}$ ; denote  $\mathbf{a}^{(j)}$  as column  $j = 1, \dots, n$  of  $\mathbf{A}$
- 4 Calculate sigma points:

$$\begin{aligned} \tilde{\mathbf{x}}_{i-1}^{(j)} &= \mathbf{x}_{i-1} + \check{\mathbf{x}}^{(j)} \quad \text{for } j = 0, \dots, 2n & \check{\mathbf{x}}^{(0)} &= 0 \\ \check{\mathbf{x}}^{(j)} &= \mathbf{a}^{(j)} \sqrt{n + \kappa} \quad \text{for } j = 1, \dots, n & \check{\mathbf{x}}^{(n+j)} &= -\mathbf{a}^{(j)} \sqrt{n + \kappa} \quad \text{for } j = 1, \dots, n \end{aligned}$$

- 5 Propagate all sigma points using numerical integration:

$$\text{Initial condition: } \tilde{\mathbf{x}}_{i-1}^{(j)} \quad \text{Differential eq: } \dot{\mathbf{x}} = f(\mathbf{x}, t) \quad \text{Integration results: } \tilde{\mathbf{x}}_i^{(j)}$$

- 6 Calculate propagated state and covariance:

$$\bar{\mathbf{x}}_i = \sum_{j=0}^{2n} w_m^{(j)} \tilde{\mathbf{x}}_i^{(j)} \quad \bar{\mathbf{P}}_i = \sum_{j=0}^{2n} w_c^{(j)} (\tilde{\mathbf{x}}_i^{(j)} - \bar{\mathbf{x}}_i)(\tilde{\mathbf{x}}_i^{(j)} - \bar{\mathbf{x}}_i)^T + \mathbf{Q}_i$$

- 7 Transform sigma points and calculate predicted observation:

$$\tilde{\mathbf{y}}_i^{(j)} = G(\tilde{\mathbf{x}}_i^{(j)}, t_i) \quad \hat{\mathbf{y}}_i = \sum_{j=0}^{2n} w_m^{(j)} \tilde{\mathbf{y}}_i^{(j)}$$

- 8 Calculate predicted observation covariance, residuals, and  $\Psi$ :

$$\begin{aligned} \mathbf{S}_i &= \sum_{j=0}^{2n} w_c^{(j)} (\tilde{\mathbf{y}}_i^{(j)} - \hat{\mathbf{y}}_i)(\tilde{\mathbf{y}}_i^{(j)} - \hat{\mathbf{y}}_i)^T + \mathbf{R}_i \\ \boldsymbol{\nu}_i &= \mathbf{y}_i - \hat{\mathbf{y}}_i & \boldsymbol{\Psi}_i &= \boldsymbol{\nu}_i^T (\mathbf{S}_i)^{-1} \boldsymbol{\nu}_i \end{aligned}$$

- 9 Use Kalman filter equations to update estimates:

$$\begin{aligned} \mathbf{V}_i &= \sum_{j=0}^{2n} w_c^{(j)} (\tilde{\mathbf{x}}_i^{(j)} - \bar{\mathbf{x}}_i)(\tilde{\mathbf{y}}_i^{(j)} - \hat{\mathbf{y}}_i)^T & \mathbf{K}_i &= \mathbf{V}_i (\mathbf{S}_i)^{-1} \\ \hat{\mathbf{x}}_i &= \bar{\mathbf{x}}_i + \mathbf{K}_i \boldsymbol{\nu}_i & \hat{\mathbf{P}}_i &= \bar{\mathbf{P}}_i - \mathbf{K}_i \mathbf{S}_i \mathbf{K}_i^T \end{aligned}$$

- 10 Return to step 1, process next observation
- 

that  $\mathbf{R} = \text{diag}(\sigma_\alpha, \sigma_\delta)$ , where  $\sigma$  represents the standard deviation of the error. The bias error is discussed separately in the calibration section of this paper. Once all sigma points are propagated, each is transformed into a predicted observation through the function  $G$ .

Step 7 in Algorithm 1, the propagated sigma point states are transformed from equinoctial elements into an anticipated observation vector  $\hat{\mathbf{y}}$  by leveraging Analytical Graphics, Inc.'s System Tool Kit <sup>®</sup>. The residual,  $\boldsymbol{\nu}$ , is defined as the difference between the observed and the calculated observation. Since the posterior probability distribution function (pdf) covariance update,  $\hat{\mathbf{P}}$ , within a Kalman filter is independent of the residual, it is important to assess the relationship between the covariance and residual. Since the matrix  $\mathbf{S}$  represents the predicted observation covariance,  $\Psi$  defines the squared Mahalanobis distance of the observation and indicates the accuracy of the orbital fit [16, 101, 102]. The remaining equations in Algorithm 1 follow a traditional Kalman filter and result in an updated estimate for the state,  $\hat{\mathbf{x}}$ , and covariance,  $\hat{\mathbf{P}}$ .

## C.2 Starting Covariance

To implement the UKF, a starting covariance is needed to accompany the TLE of interest. Since the candidate RSO is in a GEO orbit and observed frequently, a high accuracy covariance as proposed by Horwood is used [73]. Define the starting covariance in equinoctial elements such that  $\mathbf{P}_0 = \mathbf{A}^T \mathbf{A}$ , where

$$\mathbf{A} = \text{diag}(0.05 \text{ km}, 10^{-5}, 10^{-5}, 10^{-5}, 10^{-5}, 97 \times 10^{-6} \text{ rad}) \quad (\text{C.4})$$

The initial state is pulled directly from the published TLE.

## C.3 Data Sets

For the analysis in Chapter 4, the comparison of orbital data generated using three subsets of the data was required. The first set of data is created strictly from the control group, to provide a baseline analysis. The second data set generates an orbit by excluding data from the night with the simulated unbounded streak. The second data set provide estimated orbital positions assuming that the captured unbounded streak was thrown out, and the orbit was determined using only the two nights of data. The final data set incorporates the unbounded streak data. These data sets allow for analysis of the usefulness



of the position data in the unbounded streak. The impact of using the unbounded streak vs. ignoring the streak and using fewer observations is discussed below.

## Appendix D: Useful Tools

### D.1 Software

#### *D.1.1 MATLAB*

The chief advantage of MATLAB is its ability to execute high-level matrix algebra and built in data processing routines [103]. This benefit makes MATLAB a very capable image processing tool. It is capable of performing complex matrix calculations as well as output plots, images, and tables of data [103].

The built in image processing tool suite is capable of analyzing fits files and extracting information such as light intensity, pixel location, Gaussian spread functions, etc. Currently this technique is the most promising image processing tool used for this research and will continue to be used in the absence of better alternatives.

Vallado also translated his astrodynamics algorithms from fortran to MATLAB [8]. As this research hopes to demonstrate the applicability of processing external data for astrodynamics application Vallado's work may prove a suitable tool for the orbital determination and updates.

#### *D.1.2 Smithsonian Astrophysical Observatory Image ds9*

Smithsonian Astrophysical Observatory Image ds9 (ds9) is a powerful, open source tool available for Windows, Linux, and Macs developed by astronomers at Harvard. ds9 allows scientists to make quantitative and valid measurements from fits files. It can handle every type of fits files but prefers images; if used for spectra most astronomers will use other tools in conjunction. ds9 is also widely used by astronomers worldwide to manipulate images [22].

Image World Coordinate System Setting Utility (imwcs) is a program written by Smithsonian Astrophysical Observatory (SAO) to work within ds9 to automatically overlay a grid of  $\alpha$  and  $\delta$  onto a fits image. This technique is accomplished by embedding the WCS

from a star catalog into the fits file header. After the WCS has been written to the file any fits processing tool can be used to process the image and extract the  $\alpha$  and  $\delta$  for the satellite streak. Unfortunately this program only runs in an Image Reduction and Analysis Facility (IRAF) environment limiting the types of computers that can use it.

#### ***D.1.3 Teletrak***

AFIT's teletrak system was originally conceived to determine the orbit of LEO satellites using commercial telescopes. It contains a MATLAB Graphical User Interface (GUI) interface. Once setup the GUI displays a comprehensive star map for the location of the telescope used along with brightness predictions and satellite targeting information [11].

This targeting information can be used to predict generated satellite streaks within the telescope's FOV. The only potential drawback is that some modification would be required to get the GUI to behave as any telescope not already included in the teletrak network [11].

#### ***D.1.4 Systems Tool Kit***

Analytical Graphics Inc. (AGI)'s STK suite is designed to simulate space systems. Ground sites can be created, including sensor FOV, and contact reports generated for all satellites with public TLE's, reporting azimuth and elevation for all times the vehicle has line of sight. Combined with the ability to calculate sun angles, a script can be written to effectively simulate satellite streaks across a telescope [104].

#### ***D.1.5 Stellarium and KStars***

Stellarium and KStars are software suites designed to allow a user to use a home Personal Computer (PC) as a virtual planetarium. They calculate the position of astronomical objects and events and simulate the sky from a user defined location. They can serve as both an educational tool as well as observational aids for astronomers. As an observational aid either KStars or Stellarium can be setup to imitate a telescope system as long as the user provides: resolution x and y pixel count, the chip width and height in mm,

and the native focal length. Along with the value of the pixel width and height in microns, the multiplier of any lens used, and the diameter of the telescope used [86, 105].

Some of the advantages of Stellarium include automated access to outside sources, such as celestrack, to upload the publicly available TLE. Using this TLE it can determine if a satellite is within the FOV of the telescope, the sun angle, and its visual magnitude to determine, and automatically report, if the RSO is visible to the observer [86].

Stellarium also contains an Application Programming Interface (API) scripting interface allowing it to be plugged into a larger modeling tool, such as Phoenix Integration's Model Center, to simulate a single node in a larger sensor network. This scripting language can also be used with the above capabilities to model what a satellite streak on a CCD by scripting the sensor characteristics and exposure time to report any visible satellites  $\alpha$  and  $\delta$  in that time frame [86].

Conversely KStars allows for the creation of artificial satellites, similar to Stellarium, using TLE data but it is unclear from the documentation if it includes a sun angle calculation to determine visibility through the telescope [105].

## **D.2 Hardware**

### ***D.2.1 Meade Instruments***

Several of the telescopes used at AFIT are manufactured by Meade Instrument Corporation, including both a 10 inch and 16 inch telescope, currently mounted on Meade Autostar II mounts. These mounts allow an operator to align the optical system using one or two stars contained within the internal catalog, point the telescope at a desired point in the sky, and track an object [106]. Past work at AFIT has created programs to adjust the tracking rate of the scope so that it will not only track at the rate of the celestial sphere, useful for astrometric observations, but can also track a LEO satellite as it passes overhead [11].

## Bibliography

- [1] P. C. Saunders and C. D. Lutes, "China's ASAT test: motivations and implications," tech. rep., DTIC Document, 2007.
- [2] T. Kelso *et al.*, "Analysis of the Iridium 33 Cosmos 2251 Collision," *Advanced Maui Optical and Space Surveillance Technologies Conference Proceedings*, 2009.
- [3] B. Obama, *National Space Policy of the United States of America*. Executive Office of the President, 2010.
- [4] *Increased space, cyber threats top concerns for Air Force Space Command*. Air Force Public Affairs, 22 November 2013. <http://www.afspc.af.mil/news/story.asp?id=123371861>, Accessed 15 Dec 13.
- [5] S. McLean, "Personnal Correspondence," 14 August 2013.
- [6] "Continuing Kepler's Quest-Assessing Air Force Space Command's Astrodynamics Standards," tech. rep., National Research Council of the National Academies, 2012.
- [7] C. Sabol, A. Segerman, A. Hoskins, and et.al., "Automated Uncorrelated Track Resolution with the Search and Determination Integrated Environment (SADIE)," tech. rep., AFRL, Directed Energy Directorate, 2012.
- [8] D. A. Vallado, *Fundamentals of astrodynamics and applications*, Vol. 12. Springer, 2001.
- [9] W. E. Wiesel, *Modern Orbit Determination*. Aphelion Press, 2003.
- [10] "Space Surveillance Network Site Information Handbook," tech. rep., Air Force Space Command, 2007.
- [11] M. M. Schmunk, "Initial Determination of Low Earth Orbits Using Commercial Telescopes," tech. rep., Air Force Institute of Technology M.S. Thesis, 2008.
- [12] M. Graff, "Development of A Remotely Operated Autonomous Satellite Tracking System," tech. rep., Air Force Institute of Technology M.S. Thesis, 2010.
- [13] G. Briggs, "Satellite Detection and Real-Time Orbit Estimation with Commercial Telescopes," tech. rep., Air Force Institute of Technology M.S. Thesis, 2011.
- [14] P. R. Escobal, *Methods of orbit determination*, Vol. 1. New York: Wiley, 1965.
- [15] G. M. Goff, J. Black, and J. A. Beck, "Orbit Estimation Of A Continuously Thrusting Satellite Using Variable Dimension Filters," *AIAA SciTech*, No. 93, American Institute of Aeronautics and Astronautics, Jan 2015. doi: 10.2514/6.2015-0093.

- [16] G. M. Goff, D. Showalter, J. T. Black, and J. A. Beck, "Parameter Requirements for Noncooperative Satellite Maneuver Reconstruction Using Adaptive Filters," *Journal of Guidance, Control, and Dynamics*, Jan 2015. doi: 10.2514/1.G000941.
- [17] *Introduction to Space Activities*. Space Foundation, 2011. [www.spacefoundation.org](http://www.spacefoundation.org), Accessed 14 Aug 13.
- [18] E. M. Soop, *Handbook of geostationary orbits*, Vol. 3. Springer, 1994.
- [19] W. L. Taber and I. M. Underwood, "NASA Jet Propulsion Laboratory spice toolkit documentation function J1900," [naif.jpl.nasa.gov/pub/naif/toolkit\\_docs/FORTRAN/spicelib/j1900.html](http://naif.jpl.nasa.gov/pub/naif/toolkit_docs/FORTRAN/spicelib/j1900.html), Accessed 2 April 2014.
- [20] K. T. Alfriend and J.-I. Lim, "Performance of a Dynamic Algorithm For Processing Uncorrelated Tracks," *Lincoln Laboratory*, 1999, p. 17.
- [21] National Aeronautics and Space Administration Infrared Processing and Analysis Center Teacher Archive Research Program (NITARP), "FITS format," 13 May 2011. [http://coolwiki.ipac.caltech.edu/index.php/FITS\\_format](http://coolwiki.ipac.caltech.edu/index.php/FITS_format), Accessed 14 Dec 13.
- [22] L. Rebull, "NASA Infrared Processing and Analysis Center Teacher Archive Research Program Tutorial Episode 2: ds9: Part 1," 29 Jan 2013. <http://www.youtube.com/watch?v=C8QBwrKbEtc>, Accessed 14 Dec 13.
- [23] M. Gruss, "Shelton Orders Shutdown of Space Fence," 6 Aug 2013. <http://www.spacenews.com/article/military-space/36655shelton-orders-shutdown-of-space-fence>, Accessed 27 Aug 1.
- [24] B. Weeden, P. Cefola, and J. Sankaran, "Global Space Situational Awareness Sensors," *AMOS Conference*, 2010.
- [25] P. W. Schumacher, "Personnal Correspondence," 15 Jan 2015.
- [26] *Mathematical Foundation for SCC Astrodynamic Theory*. SPADOC Computation Center, Headquarters North American Aerospace Defense Command (ADCOM), Peterson AFB, 1980. NORAD Technical Publication TP-SCC-008.
- [27] P. Merritt, *Beam control for laser systems*. Directed Energy Professional Society, Albuquerque, NM, 2012.
- [28] S. T. Fiorino, R. J. Bartell, M. J. Krizo, G. L. Caylor, K. P. Moore, T. R. Harris, and S. J. Cusumano, "A first principles atmospheric propagation & characterization tool: the laser environmental effects definition and reference (LEEDR)," *Lasers and Applications in Science and Engineering*, International Society for Optics and Photonics, 2008, pp. 68780B–68780B.

- [29] D. L. Nishimoto, J. L. Africano, P. F. Sydney, K. M. Hamada, V. S. Hoo, P. W. Kervin, and E. G. Stansbery, "Spectroscopic observations of space objects and phenomena using Spica and Kala at AMOS," *International Symposium on Optical Science and Technology*, International Society for Optics and Photonics, 2001, pp. 212–220.
- [30] R. Lemos, "Giant camera tracks asteroids," November 2008. <http://www.technologyreview.com/news/411235/giant-camera-tracks-asteroids/>, Accessed: 18 Sep 2013.
- [31] D. Shultz, "Unpublished Paper," 26 Nov 2013.
- [32] M. Born and E. Wolf, *Principles of Optics 7th (expanded) edition*. Cambridge University Press, 2013. pg. 163-164.
- [33] R. Berry and J. Burnell, *The handbook of astronomical image processing*. Willmann-Bell, Inc, 2011.
- [34] Q. M. Lam and J. L. Crassidis, "Precision attitude determination using a multiple model adaptive estimation scheme," *Aerospace Conference, 2007 IEEE*, IEEE, 2007, pp. 1–20.
- [35] T. Sato, T. Wakayama, T. Tanaka, K.-i. Ikeda, and I. Kimura, "Shape of space debris as estimated from radar cross section variations," *Journal of spacecraft and rockets*, Vol. 31, No. 4, 1994, pp. 665–670.
- [36] J. L. Walker, "Range-Doppler imaging of rotating objects," *Aerospace and Electronic Systems, IEEE Transactions on*, No. 1, 1980, pp. 23–52.
- [37] D. T. Hall, J. L. Africano, J. V. Lambert, and P. W. Kervin, "Time-resolved i-band photometry of calibration spheres and nak droplets," *Journal of Spacecraft and Rockets*, Vol. 44, No. 4, 2007, pp. 910–919.
- [38] R. Linares, M. K. Jah, J. L. Crasidis, and C. K. Nebelecky, "Space Object Shape Characterization and Tracking Using Light Curve and Angles Data," *Journal of Guidance, Control, and Dynamics*, 2013.
- [39] M. D. Lichter and S. Dubowsky, "State, shape, and parameter estimation of space objects from range images," *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, Vol. 3, IEEE, 2004, pp. 2974–2979.
- [40] M. Kaasalainen and J. Torppa, "Optimization Methods for Asteroid Lightcurve Inversion: I. Shape Determination," *Icarus*, Vol. 153, No. 1, 2001, pp. 24–36.
- [41] M.-A. A. Cauquy, M. C. Roggemann, and T. J. Schulz, "Distance-based and neural-net-based approaches for classifying satellites using spectral measurements," *Optical Engineering*, Vol. 45, No. 3, 2006, pp. 036201–036201.

- [42] V. P. Osweiler, "Covariance estimation and autocorrelation of NORAD two-line element sets," tech. rep., Air Force Institute of Technology M.S. Thesis, 2006.
- [43] J. C. Chan and D. Navarro, "Comparisons of NORAD two-line elements with INTELSAT orbital elements," *Space Debris*, Vol. 473, 2001, pp. 771–779.
- [44] C. Früh and T. Schildknecht, "Accuracy of two-line-element data for geostationary and high-eccentricity orbits," *Journal of Guidance, Control, and Dynamics*, Vol. 35, No. 5, 2012, pp. 1483–1491.
- [45] C. Sabol, K. Hill, C. McLaughlin, K. K. Luu, and M. Murai, "Relative Orbit Determination of Geosynchronous Satellites using the COWPOKE Equations," *Sixth US/Russian Space Surveillance Workshop: August 22-26, 2005, Proceedings*, Central Astronomical Observatory at Pulkovo, Russian Academy of Sciences, 2005, p. 273.
- [46] T. Schildknecht, R. Musci, and T. Flohrer, "Properties of the high area-to-mass ratio space debris population at high altitudes," *Advances in Space Research*, Vol. 41, No. 7, 2008, pp. 1039–1045.
- [47] S. Coffey, L. Healy, and H. Neal, "Applications of parallel processing to astrodynamics," *Celestial Mechanics and Dynamical Astronomy*, Vol. 66, No. 1, 1996, pp. 61–70.
- [48] M. Wasson, "Space Situational Awareness in the Joint Space Operations Center," tech. rep., DTIC Document, 2011.
- [49] S. Blacknan and A. House, "Design and analysis of modern tracking systems," *Boston, MA: Artech House*, 1999.
- [50] A. B. Poore, "Multidimensional assignment formulation of data association problems arising from multitarget and multisensor tracking," *Computational Optimization and Applications*, Vol. 3, No. 1, 1994, pp. 27–57.
- [51] O. E. Drummond, T. L. Ogle, and S. Waugh, "Metrics for evaluating track covariance consistency," *Signal and Data Processing of Small Targets*, Vol. 6699-1, 2007, pp. 669–916.
- [52] S. S. Blackman, "Multiple hypothesis tracking for multiple target tracking," *Aerospace and Electronic Systems Magazine, IEEE*, Vol. 19, No. 1, 2004, pp. 5–18.
- [53] D. Reid, "An algorithm for tracking multiple targets," *Automatic Control, IEEE Transactions on*, Vol. 24, No. 6, 1979, pp. 843–854.
- [54] R. Singer, R. Sea, and K. Housewright, "Derivation and evaluation of improved tracking filter for use in dense multitarget environments," *Information Theory, IEEE Transactions on*, Vol. 20, No. 4, 1974, pp. 423–432.



- [55] I. J. Cox and S. L. Hingorani, "An efficient implementation and evaluation of Reid's multiple hypothesis tracking algorithm for visual tracking," *Pattern Recognition, 1994. Vol. 1-Conference A: Computer Vision & Image Processing., Proceedings of the 12th IAPR International Conference on*, Vol. 1, IEEE, 1994, pp. 437–442.
- [56] Y. Ma, S. Soatto, J. Kosecka, and S. S. Sastry, *An Invitation to 3-D Vision: From Images to Geometric Models*. Springer, New York, 2004.
- [57] F. L. Pedrotti, L. S. Pedrotti, and L. S. Pedrotti, *Introduction to optics*, Vol. 2. Prentice-Hall Englewood Cliffs, 1993.
- [58] D. Lang, D. W. Hogg, K. Mierle, M. Blanton, and S. Roweis, "Astrometry. net: Blind astrometric calibration of arbitrary astronomical images," *The Astronomical Journal*, Vol. 139, No. 5, 2010, p. 1782.
- [59] B. Sease and B. Flewelling, "GEODETICA: A General Software Platform for Processing Continuous Space-Based Imagery," *AAS Space Flight Mechanics Meeting*, No. 15-409, American Astronautical Society, Jan 2015.
- [60] D. Moomey, "Aiding Geostationary Space Situational Awareness Using Small Aperture Commercial Telescopes," tech. rep., Air Force Institute of Technology M.S. Thesis, 2015.
- [61] F. Oniga, M. Miron, R. Danescu, and S. Nedevschi, "Automatic recognition of low earth orbit objects from image sequences," *Intelligent Computer Communication and Processing (ICCP), 2011 IEEE International Conference on*, IEEE, 2011, pp. 335–338.
- [62] B. Wallace, R. Scott, and A. Spaans, "The DRDC Ottawa Space Surveillance Observatory," *Advanced Maui Optical and Space Surveillance Technologies Conference*, Vol. 1, 2007, p. 2.
- [63] B. Sease and B. Flewelling, "Polar and Spherical Image Transformations for Star Localization and RSO Discrimination," *AAS/AIAA Space flight mechanics specialist conference and exhibit*, Williamsburg VA, AAS, 2015. AAS 15-412.
- [64] C. C. Liebe, K. Gromov, and D. M. Meller, "Toward a stellar gyroscope for spacecraft attitude determination," *Journal of Guidance, Control, and Dynamics*, Vol. 27, No. 1, 2004, pp. 91–99.
- [65] M. Levesque, "Automatic reacquisition of satellite positions by detecting their expected streaks in astronomical images," *Advanced Maui Optical and Space Surveillance Technologies Conference*, Vol. 1, 2009, p. 81.
- [66] S. Nikolaev, D. Phillion, L. Simms, A. Pertica, S. S. Olivier, and R. Cognion, "Analysis of Galaxy 15 Satellite Images from a Small-Aperture Telescope," tech. rep., DTIC Document, 2011.

- [67] C. Sabol, "Search and Determine Integrated Environment: Enabling Threat Indications and Warning through Automated UCT Resolution," Presentation, Wright Patterson AFB, 31 July 2012).
- [68] M. Jah, "Personnal Correspondence," 14 August 2013-Present.
- [69] S. L. Coffey, E. Jenkins, H. L. Neal, and H. Reynolds, "Parallel processing of uncorrelated observations into satellite orbits," *Spaceflight mechanics 1996*, 1996, pp. 739–756.
- [70] B. Tapley, B. Schutz, and G. H. Born, *Statistical Orbit Determination*. Burlington, MA: Elsevier, 2004.
- [71] S. J. Julier and J. K. Uhlmann, "Reduced Sigma Point Filters for the Propagation of Means and Covariances through Nonlinear Transformations," *American Control Conference, Proceedings of the*, Vol. 2, Institute of Electrical and Electronics Engineers, 2002, pp. 887–892. doi: 10.1109/ACC.2002.1023128.
- [72] S. J. Julier and J. K. Uhlmann, "Unscented Filtering and Nonlinear Estimation," *Proceedings of the IEEE*, Vol. 92, Mar 2004, pp. 401–422. doi: 10.1109/JPROC.2003.823141.
- [73] J. T. Horwood, J. M. Aristoff, N. Singh, and A. B. Poore, "A Comparative Study of New Non-linear Uncertainty Propagation Methods for Space Surveillance," *Signal and Data Processing of Small Targets, Proceedings 9092*, No. 90920H, International Society for Optics and Photonics, Jun 2014. doi: 10.1117/12.2051353.
- [74] P. F. Sydney, J. L. Africano, A. Fredericks, K. M. Hamada, V. S. Hoo, D. L. Nishimoto, P. W. Kervin, S. Bisque, and M. Bisque, "Raven automated small telescope systems," *International Symposium on Optical Science and Technology*, International Society for Optics and Photonics, 2000, pp. 237–247.
- [75] M. L. Thrall, "Orbit determination of highly eccentric orbits using a RAVEN telescope," tech. rep., DTIC Document, 2005.
- [76] M. Bolden, "Personnal Correspondence," 23 September 2013.
- [77] B. Handwerk, "Worlds largest digital camera to watch for killer asteroids," June 2010. <http://news.nationalgeographic.com/news/2010/06/100622-science-space-digital-camera-asteroid-telescope>, Accessed: 18 Sep 2013.
- [78] J. Tyson, "Large Synoptic Survey Telescope," *Proc. SPIE*, Vol. 4836, 2003, pp. 10–20.
- [79] R. D. Richmond and S. C. Cain, *Direct-detection LADAR systems*. SPIE Press, 2010. pg. 74-75.

- [80] J. W. Goodman, *Introduction to Fourier optics*. Roberts and Company Publishers, 2005. pg. 67-75, 101.
- [81] D. G. Voelz, *Computational Fourier Optics*. SPIE Press, 2011. pg. 141-168.
- [82] P. H. Merritt, *Beam Control for Laser Systems*. Directed Energy Professional Society, High Energy Laser Joint Technology Office, 2012.
- [83] F. R. Hoots and R. L. Roehrich, "Models for propagation of NORAD element sets.," 1980.
- [84] D. A. Vallado, "An Analysis of State Vector Prediction Accuracy," *Paper USR*, 2007, pp. 07–S6.
- [85] A. Klaus, J. Bauer, K. Karner, P. Elbischger, R. Perko, and H. Bischof, "Camera calibration from a single night sky image," *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, Vol. 1, IEEE, 2004, pp. I–151.
- [86] M. Gates, "Stellarium User Guide," 4 October 2013. [http://www.stellarium.org/wiki/index.php/Stellarium\\_User\\_Guide](http://www.stellarium.org/wiki/index.php/Stellarium_User_Guide), Accessed 14 Dec 13.
- [87] W. E. Wiesel, *Spaceflight Dynamics*. Irwin McGraw-Hill, 1997.
- [88] M. E. Liggins, D. D. L. Hall, and J. Llinas, *Handbook of multisensor data fusion: theory and practice*. CRC press, 2008.
- [89] M. P. Levesque and S. Buteau, "Image processing technique for automatic detection of satellite streaks," tech. rep., DTIC Document, 2007.
- [90] M. Lévesque and R. Defence, *Improving satellite-streak detection by the use of false alarm rejection algorithms*. Defence R&D Canada-Valcartier, 2008.
- [91] G. M. Goff, J. Black, and J. Beck, "Parameter Requirements for Non-Cooperative Satellite Maneuver Reconstruction Using Adaptive Filters," *SPACE Conferences & Exposition*, American Institute of Aeronautics and Astronautics, Aug. 2014. doi: 10.2514/6.2014-4169, 10.2514/6.2014-4169.
- [92] M. Bolden, "Personnal Correspondence," 5 September 2013.
- [93] F. H. Lutze, "Astromechanics Supplemental Information 12. Satellite Look Angle," AOE 4134 Lecture, 2002.
- [94] J. M. Aristoff, J. T. Horwood, N. Singh, and A. B. Poore, "Nonlinear Uncertainty Propagation in Orbital Elements and Transformation to Cartesian Space Without Loss of Realism," *SPACE Conferences & Exposition*, No. 4167, American Institute of Aeronautics and Astronautics, Aug 2014. doi: 10.2514/6.2014-4167, 10.2514/6.2014-4167.

- [95] J. T. Horwood, J. M. Aristoff, N. Singh, A. B. Poore, and M. D. Hejduk, “Beyond Covariance Realism: A New Metric for Uncertainty Realism,” *Signal and Data Processing of Small Targets, Proceedings 9092*, No. 90920F, International Society for Optics and Photonics, Jun 2014. doi: 10.1117/12.2054268.
- [96] R. A. Broucke and P. J. Cefola, “On the Equinoctial Orbit Elements,” *Celestial Mechanics*, Vol. 5, May 1972, pp. 303–310. doi: 10.1007/bf01228432.
- [97] D. Simon, *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. Hoboken, NJ: John Wiley & Sons, 2006.
- [98] S. Julier, J. Uhlmann, and H. F. Durrant-Whyte, “A New Method for the Nonlinear Transformation of Means and Covariances in Filters and Estimators,” *Automatic Control, IEEE Transactions on*, Vol. 45, Mar 2000, pp. 477–482. doi: 10.1109/9.847726.
- [99] A. S. Ackleh, E. J. Allen, R. B. Kearfott, and P. Seshaiyer, *Classical and Modern Numerical Analysis: Theory, Methods and Practice*. Boca Raton, FL: CRC Press, 2011.
- [100] Microcosm Inc., “High Precision Orbit Propagator,” Jan 2015. <http://smad.com/space-systems/software/high-precision-orbit-propagator/>, Accessed 19 Jan 2015.
- [101] Y. Bar-Shalom and K. Birmiwal, “Variable Dimension Filter for Maneuvering Target Tracking,” *Aerospace and Electronic Systems, IEEE Transactions on*, Vol. 18, Sep 1982, pp. 621–629. doi: 10.1109/TAES.1982.309274.
- [102] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation: Theory Algorithms and Software*. Hoboken, NJ: John Wiley & Sons, 2004.
- [103] S. Kay, *Intuitive probability and random processes using MATLAB®*. Springer, 2006.
- [104] A. G. Inc, “STK Fundamentals, Version 9,” 21 December 2011. [www.agi.com](http://www.agi.com), Accessed 14 Dec 13.
- [105] J. Harris, “The KStars Handbook,” <http://docs.kde.org/stable/en/kdeedu/kstars/kstars.pdf>, Accessed 14 Dec 13.
- [106] M. I. Corperation, “Instruction Manual: 8”, 10”, 12”, 14”, 16”, LX200GPS Schmidt-Cassegrain Telescopes with Autostar II Hand Controller,” 2003.

REPORT DOCUMENTATION PAGE					Form Approved OMB No. 0704-0188	
<p>The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.</p>						
1. REPORT DATE (DD-MM-YYYY)		2. REPORT TYPE		3. DATES COVERED (From — To)		
17-09-2015		Doctoral Dissertation		Oct 2012-Sep 2015		
4. TITLE AND SUBTITLE  Leveraging External Sensor Data for Enhanced Space Situational Awareness				5a. CONTRACT NUMBER		
				5b. GRANT NUMBER		
				5c. PROGRAM ELEMENT NUMBER		
6. AUTHOR(S)  Bellows, Charlie T., Major, USAF				5d. PROJECT NUMBER		
				5e. TASK NUMBER		
				5f. WORK UNIT NUMBER		
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 Hobson Way WPAFB, OH 45433-7765				8. PERFORMING ORGANIZATION REPORT NUMBER  AFIT-ENY-DS-15-S-054		
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Timothy Hauenstein, 1st Lt, USAF Center for Rapid Innovation Air Force Research Laboratory 1451 Fourth St Kirtland AFB NM 87116 DSN 263-3123 Comm: (505)-853-3123				10. SPONSOR/MONITOR'S ACRONYM(S)  AFRL/RV		
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)		
12. DISTRIBUTION / AVAILABILITY STATEMENT DISTRIBUTION STATEMENT A: APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED						
13. SUPPLEMENTARY NOTES This work is declared a work of the U.S. Government and is not subject to copyright protection in the United States.						
14. ABSTRACT Reliable SSA is a recognized requirement in the current congested, contested, and competitive environment of space operations. A shortage of available sensors and reliable data sources are some current limiting factors for maintaining SSA. Unfortunately, cost constraints prohibit drastically increasing the sensor inventory. Alternative methods are sought to enhance current SSA, including utilizing non-traditional data sources (external sensors) to perform basic SSA catalog maintenance functions. Astronomical data, for example, routinely collects serendipitous satellite streaks in the course of observing deep space; but tactics, techniques, and procedures designed to glean useful information from those collects have yet to be rigorously developed. This work examines the feasibility and utility of performing ephemeris positional updates for a RSO catalog using metric data obtained from RSO streaks gathered by astronomical telescopes. The focus of this work is on processing data from three possible streak categories: streaks that only enter, only exit, or cross completely through the astronomical image. Successful use of this data will aid in resolving uncorrelated tracks, space object identification, and threat detection. Incorporation of external data sources will also reduce the number of routine collects required by existing SSA sensors, freeing them up for more demanding tasks. The results clearly demonstrate that accurate orbital reconstruction can be performed using an RSO streak in a distorted image, without applying calibration frames and that partially bound streaks provide similar results to traditional data, with a mean degradation of 6.2% in right ascension and 42.69% in declination. The methodology developed can also be applied to dedicated SSA sensors to extract data from serendipitous streaks gathered while observing other RSOs.						
15. SUBJECT TERMS Space Situational Awareness, Satellite Streaks, Astrodynamics, Orbit Determination						
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON	
a. REPORT	b. ABSTRACT	c. THIS PAGE			Dr. Jonathan Black (ENY)	
U	U	U	UU	165	19b. TELEPHONE NUMBER (include area code) (540) 231-0037 jonathan.black@vt.edu	